ABSTRACT

Title of dissertation: Scene and Video Understanding
Arpit Jain, Doctor of Philosophy, 2014

Dissertation directed by: Professor Larry S. Davis
Department of Electrical and Computer Engineering

There have been significant improvements in the accuracy of scene understanding due to a shift from recognizing objects "in isolation" to context based recognition systems. Such systems improve recognition rates by augmenting appearance based models of individual objects with contextual information based on pairwise relationships between objects. These pairwise relations incorporate common sense world knowledge such as co-occurrences and spatial arrangements of objects, temporal consistency, scene layout, etc. However, these relations, even though consistent in the 3D world, change due to viewpoint of the scene. In this thesis, we investigate incorporating contextual information from three different perspectives for scene and video understanding (a) "what" contextual relations are useful and "how" they should be incorporated into Markov network during inference, (b) jointly solving the segmentation and recognition problem using a multiple segmentation framework based on contextual information in conjunction with appearance matching, and (c) proposing a discriminative spatio-temporal patch based representation for videos which incorporates contextual information for video understanding.

Our work departs from traditional view of incorporating context into scene understanding where a fixed model for context is learned. We argue that context is

scene dependent and propose a data-driven approach to predict the importance of relationships and construct a Markov network for image analysis based on statistical models of global and local image features. Since all contextual information is not equally important, we also address the related problem of predicting the feature weights associated with each edge of a Markov network for evaluation of context. We then address the problem of fixed segmentation while modeling context by using a multiple segmentation framework and formulating the problem as "a jigsaw puzzle". We formulate the labeling problem as segment selection from a pool of segments (jigsaws), assigning each selected segment a class label. Previous multiple segmentation approaches used local appearance matching to select segments in a greedy manner. In contrast, our approach is based on a cost function that combines contextual information with appearance matching. A relaxed form of the cost function is minimized using an efficient quadratic programming solver.

Lastly, we propose a new representation for videos based on mid-level discriminative spatio-temporal patches. These patches might correspond to a primitive human action, a semantic object, or perhaps a random but informative spatiotemporal patch in the video. What define these spatiotemporal patches are their discriminative and representative properties. We automatically mine these patches from hundreds of training videos and experimentally demonstrate that these patches establish correspondence across videos. We propose a cost function that incorporates co-occurrence statistics and temporal context along with appearance matching to select subset of these patches for label transfer. Furthermore, these patches can be used as a discriminative vocabulary for action classification.

Scene and Video Understanding

by

Arpit Jain

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2014

Advisory Committee:
Professor Larry S. Davis, Chair/Advisor
Professor Rama Chellappa, Dean's Representative
Professor Min Wu
Professor David Jacobs
Professor Ramani Duraiswami

Dedication

This thesis is dedicated to my family members(parents, brother and wife), teachers and friends for their unconditional support and guidance during the course of my PhD.

Acknowledgments

I owe my gratitude to all the people who have made this thesis possible and who made my journey through graduate studies a memorable one.

First and foremost I'd like to thank my advisor Professor Larry Davis for giving me the opportunity to work with him and his constant guidance and support over the course of my PhD degree. He gave me the flexibility and freedom to pursue research topics of my interests and was always accessible to answer my questions. His immense knowledge of the field and deep understanding of problems gave me right direction throughout my research work. This thesis wouldn't have been possible without him.

I would like to give special thanks to Dr. Abhinav Gupta, former student of Dr. Davis and now a research professor at Carnegie Mellon University, who guided and mentored me as a senior during the course of my PhD. He patiently listened to my questions, explained me nitty-gritty of doing research, discussed problems and helped me in difficulties during my research.

My sincere thanks to my teachers Professor Rama Chellappa, Professor David Jocobs and Professor Min Wu who taught me vision related courses. The material covered in these courses laid the foundation of my research. I thoroughly enjoyed the class discussions, paper presentations and homework as part of the coursework. I am also grateful to Professor Ramani Duraiswami for agreeing to serve on my thesis committee.

I would like to thank all my colleagues at computer vision laboratory (Stephen,

Choi, Guangxiao, Brandyn, HyungTae, Dr. Yang, Sravanthi, Ejaz, Sameh, Mohammad, Jaishankar, Jayant and others) for enlivening my graduate life experience both at school and outside it. I would like to sincerely thank Dr. Behjat, Dr. Vlad and Dr. Zhuolin for proof reading paper manuscripts, giving their valuable feedback on my research progress and helping me with coding troubleshoots.

I owe my deepest thanks to my family - my mother, father, wife and brother who have always stood by me and have pulled me through low moments of my life. My special thanks to my wife Navita who supported, encouraged and showed remarkable patience with me especially during conference and project deadlines. Words cannot express the gratitude I owe her.

I'd like to express my gratitude to my roommates Ratnesh Tiwari, Bhaven Mehta, Rangarajan Padmanabhan, Bhupal Dev and Anand Gupta for their friendship, support and keeping the home a welcoming place.

Lastly, I thank Almighty God, who as supersoul dwells in everyone's heart and destroy with the shining lamp of knowledge the darkness born out of ignorance.

Table of Contents

List of Figures

Chapter 1: Introduction

Scene understanding is one of the central objectives of computer vision. Any autonomous system such as a robot or vehicle needs to understand the environment it is placed in and be able to make decisions based on understanding of it. Success of these systems will depend a lot on how well they are able to parse the scene, identify the objects present in the scene and make logical decisions in both normal and adverse situations. Scene understanding, at system level, involves segmenting the scene into meaningful regions, recognizing different objects present in the scene and understanding the relationships between different elements of the scene. Each of these problems has received considerable attention for images and videos within the computer vision community.

Our goal is to create representations of the world depicted in images and videos based on physical and causal relationships. These relationships encode the semantic knowledge and serve as contextual information between objects in the scene. There have been significant improvements in the performance of object recognition systems due to a shift from recognizing objects in isolation to context based recognition systems. In this thesis, we investigate problems associated with modeling context for scene understanding.

1.1 Context for Scene Understanding

There is a broad agreement in the vision community on the importance of context in scene understanding. [2–9] showed improved performance for recognition task by incorporating context along with modeling appearance of objects. Contextual information can be gathered from multiple sources. Most common among them are (1) local pixel information - patch around the region of interest (2) semantic context - object co-occurrence, scene category, spatial relations among objects (3) 2D scene context - global image statistics (4) 3D scene context - scene layout, surface orientations, support surface, horizon line (5) Temporal context - temporally proximal images, videos of similar scenes, time of capture (6) Illumination context - sun direction, sky color, shadow contrast. Some not so commonly used sources for context include geographic context - GPS location, population density, and cultural context - photographer bias, data selection bias, etc.

1.1.1 Scene dependent contextual modeling

"Is all contextual information equally important?", "Can the relation between two objects be highly uncertain depending on the scene?" and "Will it help to use context in case the relations are highly uncertain?". These are some pertinent questions which arise when modeling context. Progress towards answering these questions represents many of the contributions of this thesis. Most previous approaches treated all contextual information between pair of objects equally for image analysis. We show that modeling context in a scene dependent manner can improve

the quality of scene labeling. We describe a data-driven approach which evaluates the importance of specific contextual relations based on overall scene statistics; the approach essentially learns which contextual elements contribute to correct labeling as a function of the statistics of the scene. Experimental results indicate that this scene dependent context selection model improves performance over fully-connected and neighborhood connected Markov networks.

1.1.2 Incorporating context in a multiple segmentation and recognition framework

We also study the problem of incorporating context into a multiple segmentation framework for recognition. Segmentation algorithms based on color and texture information generally tend to fragment natural objects such as car, airplane, building, etc which have complex internal appearance variations into multiple segments. Since recognition models are learned on training data in which objects are individually segmented, performance drops when these models are applied to real segmentation. A multiple segmentation framework overcomes the issue of fragmentation by generating a large pool of segments based on computing many alternative segmentations - either by varying parameters of a segmentation algorithm or by merging segments into larger regions. The goal then is to select a subset of segments based on their matches to semantic classes which best explain the scene. However, previous works in the multiple segmentation framework [10, 11] did not include context during selection process.

We present an approach to incorporate contextual information in a multiple segmentation framework. Our cost function models the appearance and contextual information together and is efficiently solved using a Quadratic Programming framework. One of the assumptions of the multiple segmentation framework is that a good segment representing objects is present among the pool of segments. This assumption doesn't hold well in the case of natural objects. For example, window and bodies of car are never segmented as single region. Therefore, we also propose a merging function that merges heterogeneous segments based on edge, color and texture statistics.

1.2 Video Representation and Classification

Video Understanding is critical for many applications in computer vision such as surveillance, autonomous navigation, content based retrieval systems and traffic planning. In this thesis, we try to answer a basic question pertaining to video processing - "How should video be represented?" Representation determines the scope of information that can be retrieved from videos. A global representation of a video can at best help in classification task. On the other hand, representing videos based on semantics - nouns and verbs - allows us to not only classify videos but also establish strong correspondence as to when an action is performed, who performed that action and the duration of a particular action. However, current object and action detectors are not robust enough to allow this type of representation.

1.2.1 Mid-level patch based representation

We propose a discriminative patch based representation for videos in this thesis. The goal is to automatically mine spatio-temporal patches from training videos which captures the diversity of an action class and are distinct from patches of other action classes. These spatio-temporal patches might correspond to a primitive human action, a semantic object, human-object pair or perhaps a random but informative spatio-temporal patch in the video. We do not use any priors to select discriminative patches; rather we let the data select the patches. We propose an exemplar-Support Vector Machine (e-SVM) based framework to mine these patches which can be used not only for classification task but also establish strong correspondence with the test video. We propose a cost function which incorporates temporal consistency and co-occurrence statistics of these patches along with appearance matching to select a subset of these patches for label transfer which can be used for finer level action recognition and pose estimation.

1.2.2 Text detection and recognition for event detection in consumer videos

We present an end-to-end system for text detection and recognition in natural scenes and consumer videos. Text detection in natural scenes is a challenging problem compared to document text because of low contrast with background, large variation in font, color, scale and orientation combined with background clutter.

Text is also an important source of context for scene and video understanding, especially for robots or vehicles navigating in urban environments. We propose a maximally stable extremal region based framework for text detection. The detected words are passed to an Optical Character Recognition (OCR) system for recognition. We show significant improvement in text detection and recognition tasks over previous approaches on a large consumer video dataset. Furthermore, an event detection system built upon the OCR output of this approach outperformed multiple other OCR-only based submissions in the recently concluded NIST TRECVID 2013 multimedia event detection evaluations (Ek100 condition) on 100,000 videos.

1.3 Organization

The thesis is organized as follows. In Chapter 2, we present an approach to jointly solve feature weighing and edge importance in a Markov Network for applying contextual information for scene labeling. We show that our approach removes spurious edges and improves performance on standard datasets. In Chapter 3, we present an approach to jointly solve segmentation and recognition problem for scene labeling. In Chapter 4, we discuss mid-level representation for videos. We will also discuss advantages of this representation over semantic based ones. In Chapter 5, we propose a text recognition system for large scale video classification. In Chapter 6, we conclude and explore future research directions.

Chapter 2: Scene Dependent Contextual Models

2.1 Introduction

Consider the image shown in Figure 2.1, where our goal is to identify the unknown label of the region outlined in red (which we will refer to as the target), given the labels of other regions in the image. The regions labeled as building tend to force the label of the target towards building (two building regions co-occur more often than building and car) and the region labeled car tends to force the label of the target to be road, since car above road has higher probability in the contextual model than car above building. In the case of fully-connected models, the edges from the building regions to the target region outnumber the edges from other regions to the target and therefore the target is incorrectly labeled as building. If we had only utilized the relationship from the region associated with the car and ignored the relationships from other objects to predict the label of the target, then we would have labeled the target correctly.

In this proposal, we evaluate the importance of individual contextual-constraints and use a data-driven model for selection of **what** contextual constraints should be employed for solving a specific scene understanding problem, and for constructing a corresponding Markov-network. Unlike previous approaches that use fully connected

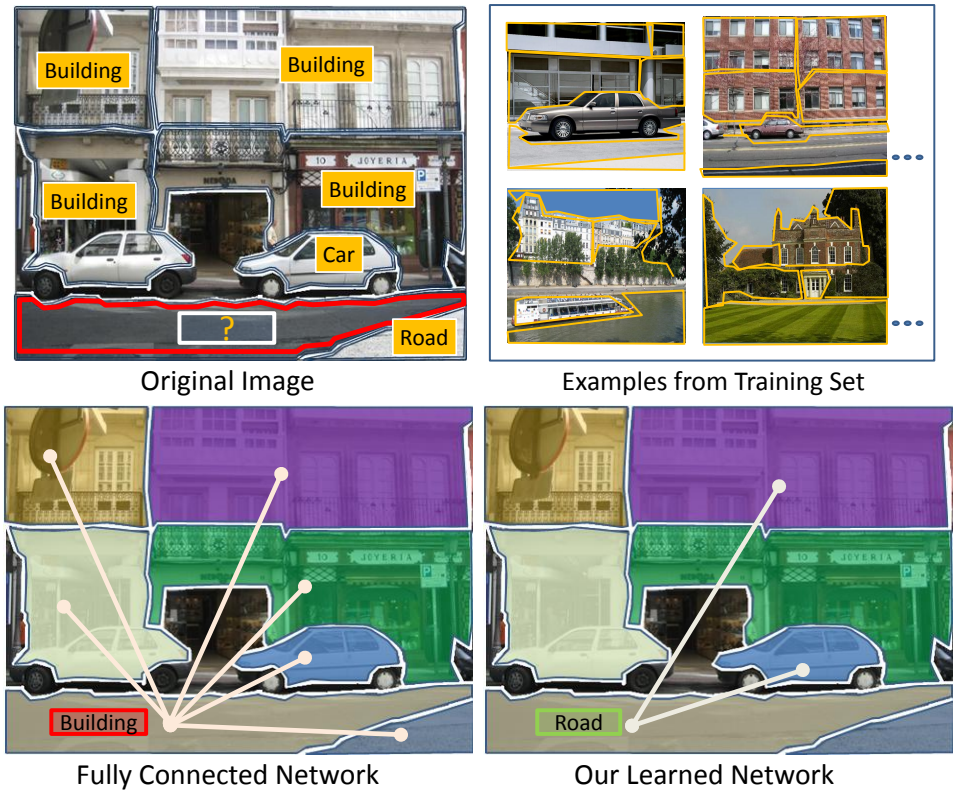


Figure 2.1: An example from our dataset showing that all relations are not informative in fully a connected network and can lead to wrong labeling and how our proposed method learns "what" edges are important and removes dubious information

or fixed structures based on neighborhood relationships, our approach predicts the structure of the Markov network (i.e., selects edges). Selection of edges is generally dependent on a combination of global and local factors such as discriminativeness of regions. However, identifying the variables/factors associated with predicting the importance of a contextual edge a priori is difficult. Instead, we take a data driven approach to predict the importance of an edge, in which scenes similar to a "test" image are identified in the training dataset and utilized to predict which regions

should be linked by an edge in the Markov network corresponding to the test image - referred to as edge prediction. Figure 2.1(a) shows an example of edge prediction from our test dataset. Our approach appropriately eliminates the edges from most of the building regions to the target and maintains the edge from the car. This leads to a correct labeling of the target.

To learn a data-driven(non-parametric) model of edge importance, we have to compute the importance of edges in the training data-set itself. This requires evaluating each edge in the training data-set with respect to other edges in the training data-set. Edges that represent consistent spatial-relationships between pairs-of-nouns are retained as informative edges and the rest are dropped. If a single 2D-spatial relationship was sufficient to represent constraints between a pair of nouns, then extracting consistent edges would be straight-forward. However, relationships between pairs of nouns are themselves scene-dependent (due to viewpoint, functional-context, etc.). For example, based on viewpoint, a road might be either below a car or around a car (see Figure 2.1(b)). Similarly, relationships are also based on function-context of an object. For example, a bottle can either be on the table or below the table based on its function (drinking vs. trash). Therefore, we cluster the relationships between pairs of nouns based on scene properties. For each cluster, we then learn feature-weights which reflect how much each feature of the vector of variables capturing spatial relationships is important for evaluating constraint/relationship satisfaction. For example, in a top-down view, road being "around" car is most important. Our approach not only learns the construction model for Markov networks, but also learns the feature weights which define how

to evaluate the degree to which a relationship between a pair of nouns is satisfied. Again, instead of explicitly modeling the factors on which these feature weights depend, we utilize a data driven approach to produce pseudo-clusters of images and estimate **how** each contextual edge should be evaluated (See Figure 2.1(b)) in each cluster.

The contributions of our work are: (1) A data driven approach for predicting **what** contextual constraints are important for labeling a specific image that uses only a subset of the relationships used in a fully-connected model. The resulting labeling are both more accurate and computed more efficiently compared to the fully-connected model. (2) A model for predicting **how** each contextual edge should be evaluated. Unlike previous approaches, which utilize a single spatial-relationship between a pair of objects (car above road), we learn a scene dependent model of context and can encode complex relationships (car above road from a standing person's viewpoint, but road around car from a top-down viewpoint).

2.2 Related Work

Recent research has shown the importance of context in many image and video understanding tasks [3, 4, 9, 12, 13]. Some of these tasks include segmentation and recognition. For object recognition, researchers have investigated various sources of context, including context from the scene [14], objects [4] and actions [15]. Scene based context harnesses global scene classification such as urban, landscape, kitchen etc to constrain the objects that can occur in the scene (for example, a car cannot

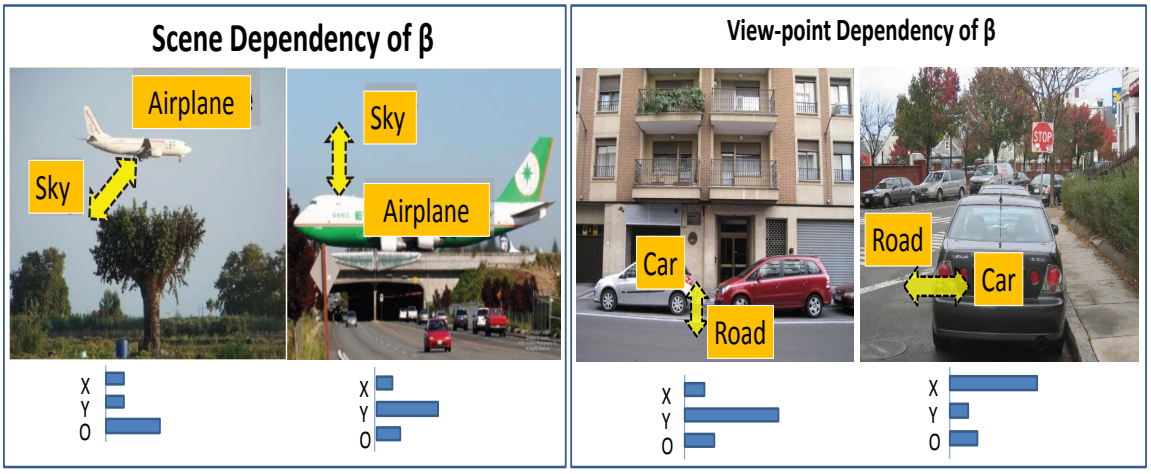


Figure 2.2: The figure shows examples of how feature weights are a function of both local and global factors. Here we show how feature weights depend on function context and viewpoint. Pairwise features X,Y and O refer to differences in x-coordinates,difference in y-coordinates and overlap between two regions respectively

occur in a kitchen). On the other hand, object based contextual approaches model object-object co-occurrence and spatial relationships to constrain the recognition problem (for example, car above road). Recent research suggests that the object-object based contextual models outperform the scene based contextual models [16]. Our work builds upon this and tries to improve how object-object relationships should be utilized on selected pairs of regions instead of all region pairs.

In most previous work, relationships are represented by graphical models such as belief networks [15] or CRFs [4], and the parameters of the graphical models are learned using graph cuts [17] or max-margin method [18]. One of the common problems with such approaches is determining "what" edges in the graphical model

should be used for inference. While fully-connected networks provide the largest number of constraints, they are hard to evaluate and also include weak edges which can sometimes lead to higher belief entropy. Fixed structure approaches, such as neighborhood based MRF's [2], are computationally less demanding but ignore vital long range constraints. Other approaches such as [19] perform approximate inference by selecting fewer edges based on object co-occurrences and discriminability. There has been some work on learning the structure of a graphical model from the training dataset itself [20]. Here, the edges are learned/inserted based on the consistency of relationships throughout the dataset. However, most of the contextual relationships are scene based and might not hold true for all scenarios. In such situations, structure-learning approaches tend to drop the informative edges, since they are not consistent throughout. Instead, we predict the relevant contextual relationships based on the scene being analyzed. In our approach, instead of learning a fixed structure from the training dataset, we learn the space of allowable structures and then predict a structure for a test image based on its global scene features and local features.

Our work is similar in spirit to "cautious" collective inference [21, 22]. Here, instead of using all relationships, the relationships which connect discriminative regions are used for initial iterations and the number of relationships used are increased with each iteration. However, the confidence in the classification of a region is itself a subtle problem and might be scene-dependent. Instead, we learn a decision model for dropping the edges/relationships based on global scene parameters and local parameters. Our work is also related to the feature/kernel weighting problem [23].

However, instead of learning weights of features/kernel for recognition problems, we select features for a constraint satisfaction problem. Therefore, the feature weights are on pairwise features and indicate "how" the edge in a Markov network should be evaluated. This is similar to [3] in which the prior on possible relationships between pairs of nouns is learned, where each relationship is based on one pair-wise feature. However, this approach keeps the priors/weights fixed for a given pair of nouns whereas in our case we learn a scene-dependent weight function.

2.3 Overview

Given a set of training images with ground truth labeling of segments, our goal is to learn a model which predicts the importance of an edge in a Markov network given the global features of the image and local features of the regions connected by that edge. We also want to learn a model of image and class-specific pairwise feature weights to evaluate contextual edges. Instead of modeling the latent factors and using a parametric approach for computing edge importance and feature-weights, we use a data-driven non-parametric approach to model these. Learning a non-parametric model of edge-importance would require computing edge importance in the ensemble of Markov networks of the set of training images. Edge importance, however, itself-depends upon feature weights; feature weights determine if contextual constraints are satisfied or not. On the other hand, the feature weights, themselves, depend on the structure of the Markov networks in the training dataset, since only the images for which nouns are (finally) linked by an edge should be evaluated to

compute the feature weights. We propose an iterative approach to these linked problems. We fix the feature weights to estimate the current edge-importance function, followed by fixing the edge-importance function to re-evaluate feature weights.

Learning. Figure 2.3 shows an overview of our iterative learning algorithm. Assume that at some iteration, we have some contextual edges in the training dataset and feature weights associated with each contextual edge. For example, in figure 2.3, out of the six occurrences of road and car, we have contextual edges in five cases with their corresponding weights. Based on the current feature weights, we first estimate how likely each edge satisfies the contextual relationship and its importance in identifying the labels of the regions. To compute the importance, we compare labeling performance with and without the edge in the Markov network. For example, in the first case the relative locations of the car and road are not coherent with other similar examples in the training dataset (the road is neither around/overlapping the car nor is it to the right of the car as in the other cases). Therefore, in this case the edge linking the car and road is not informative and the Markov network without the edge outperforms the Markov network with the edge.

After computing the importance of each edge, a few non-informative edges are eliminated. At this stage, we fix our edge importance function and utilize it to estimate the new pair-wise feature weights. For computing the new feature weights, we retrieve similar examples from the training dataset and analyze which pair-wise features are consistent throughout the set of retrieved samples. The weights of the consistent features are increased accordingly. In the example, we can see that for the images with a top-down viewpoint, the overlap feature becomes important since

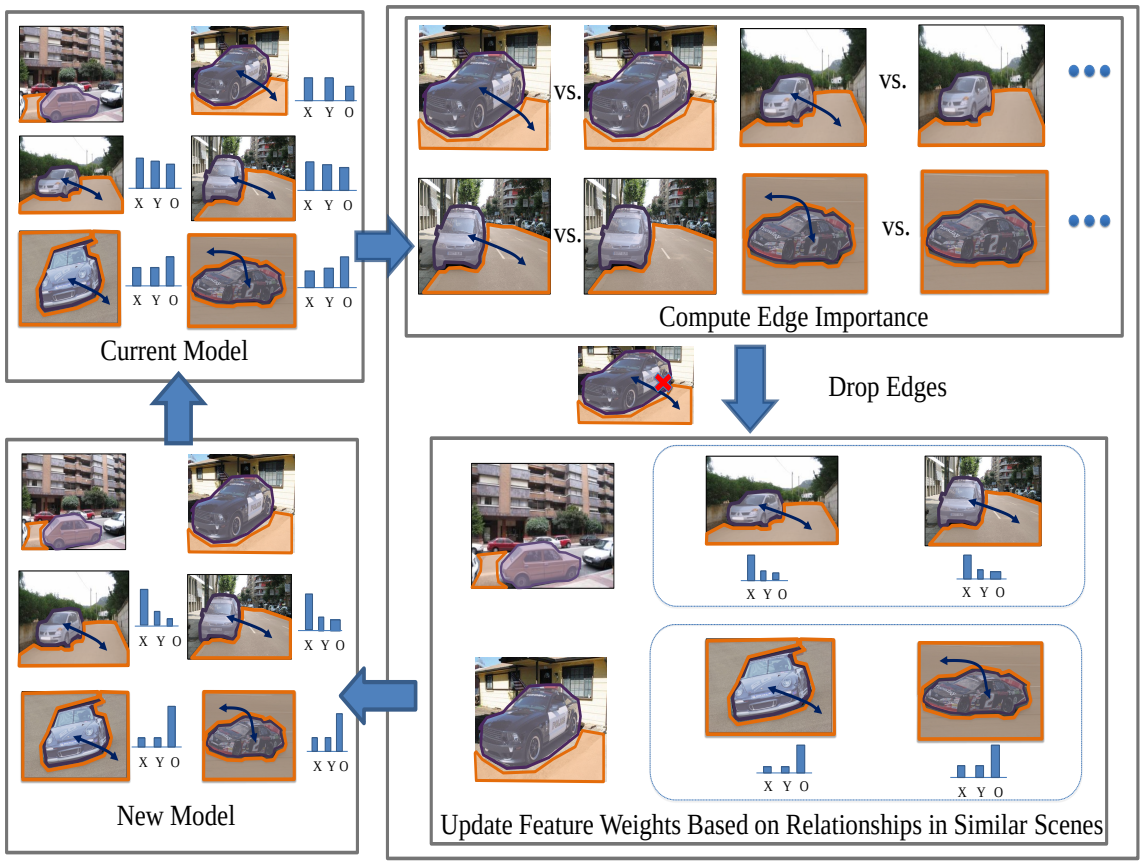


Figure 2.3: Overview of our approach: we propose an iterative approach of **what** constitutes the space of important edges and **how** these edges can be evaluated. Our approach simultaneously learn the construction model $\mathcal{F}_e()$ and differential feature weights β.

in the retrieved samples the road region was generally overlapping the car region. Once the feature weights are updated, we obtain a new non-parametric model of both edge-importance and feature weights. This new model is then used to evaluate the edge importance and drop further edges and recompute feature weights.

Inference. The inference procedure is illustrated in Figure 2.4. An image is first segmented into regions. For segmentation, we use the SWA algorithm [24]

and stability analysis for estimating the stable segmentation level [16]. We then predict the importance of each edge based on global features and local features of the regions connected by the edge. Based on the importance of edges, we construct a Markov network for inference. For each edge, we also compute feature weights that should be utilized to evaluate context on that edge. The labels are then predicted using the message passing algorithm over the constructed Markov network with the estimated feature weights.

2.4 Mathematical Formulation

We now more formally describe our approach to learn "what" edges constitute the space of efficient networks and "how" to evaluate these edges in those networks. Our motivation is that not all edges in the complete Markov network are informative. So, we want to include in our Markov network only those edges which are generally informative, given the image, and also predict the corresponding feature weights which describe how to evaluate the constraints specified by the selected edges. Formally, our goal is to learn two functions from training data: $\mathcal{F}_e(G^t, R_i^t, R_j^t)$ and $\beta(G^t, n_i^t, n_j^t)$; where $\mathcal{F}_e()$ evaluates whether there should be an edge between regions i and j of image t and $\beta()$ represents the vector of pair-wise feature weights. The function $\mathcal{F}_e()$ depends on the global scene features G^t and the local features of region i and j represented by R_i^t and R_j^t. On the other hand, the feature weights depend on the global scene features and the pair of noun classes for which the pair-wise features are being evaluated.

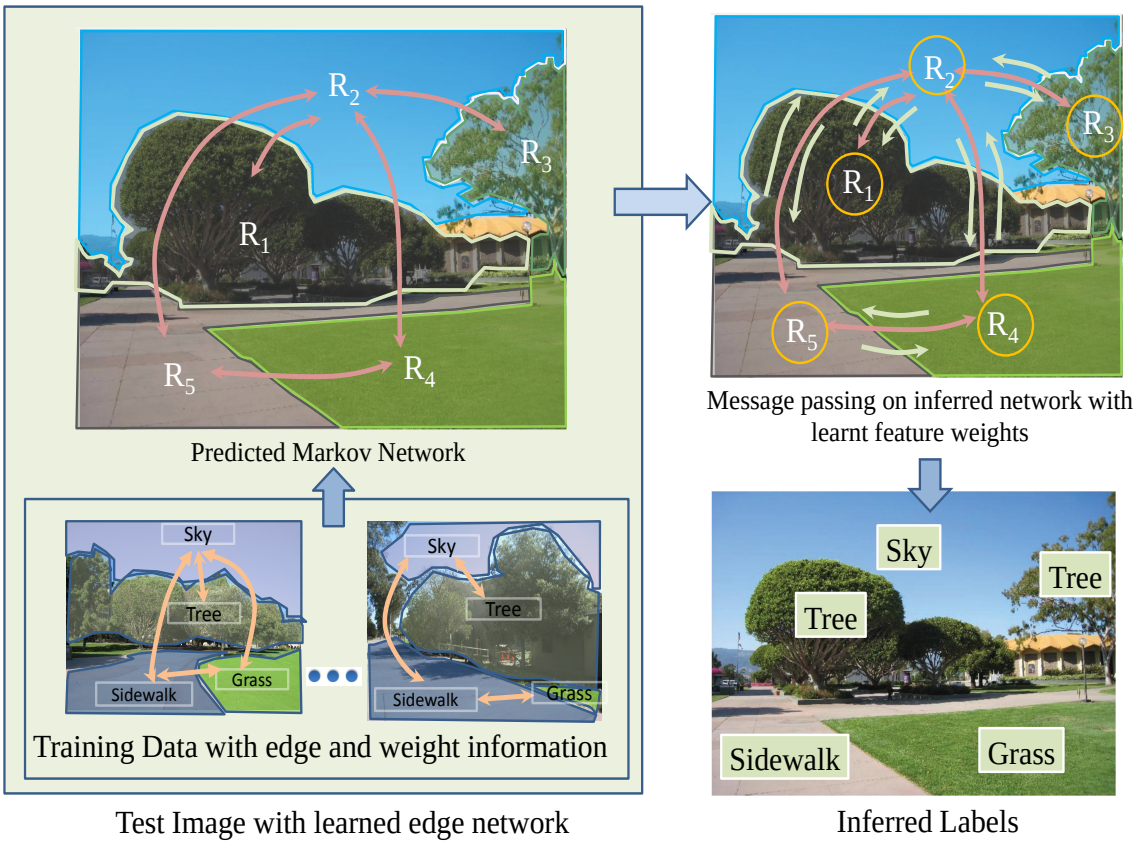


Figure 2.4: Inference algorithm for our approach: Using the global and local features computed from the segmentation, we first predict the structure of the Markov network by matching possible edges to edges in the training set (locally weighted regression). We also estimate the feature weights β for each edge in the Markov network. Finally we use message passing to predict the labels

The functions are learned based on a cost function, which minimizes the cost of labeling in the training dataset. The task is to predict all the nouns in an image, and therefore our cost function can be formulated as follows: Suppose our vocabulary consists of m object-classes, and let y_i^t be an m-dimensional vector which represents the ground-truth annotation for region i in training image t. Function $f_A(R_i)$ eval-

uates the appearance of the region by comparing it with the appearance models of the m labels and returns an m-dimensional vector with appearance distance scores. The cost function is then formulated as:

$$\mathcal{C} = \sum_{t} (\sum_{i} y_i^t f_A(R_i) + \sum_{(j,k)\in\Lambda^t} y_{jk}^t F_C(G^t, R_{jk})) \tag{2.1}$$

In this cost function, y_{jk}^t is a m^2 dimensional vector which represents pair-wise ground truth annotations and F_C is a m^2 dimensional-vector representing how well the pair-wise features R_{jk} match the contextual parameters for all m^2 pairs of labels. Λ^t represents the set of chosen edges for an image t based on function $\mathcal{F}_e$ and, therefore, we define Λ^t as: $\{(j,k) : \mathcal{F}_e(G^t, R_i^t, R_j^t) > \alpha\}$.

Contextual evaluation also requires feature-weighting, since all features are not equally important for contextual relationship evaluation. For example, while a difference in y-coordinate is important in evaluation of the contextual relationship between sky and water, the differences in x-coordinate is irrelevant. As discussed previously, these feature weights depend not only on the pair of nouns but also the global features of the scene. Therefore, if the function $f_{n_j,n_k}(G^t, R_{jk})$ represents the $(n_j, n_k)^t h$ element of F_C, we can write it as:

$$f_{n_j,n_k}(G^t, R_{jk}) = \sum_{l=1}^{L} \beta_{n_j,n_k}^l(G^t) C_{n_j,n_k}^l(G^t, R_{jk}^l) \tag{2.2}$$

where β^l represents the weight of the l^{th} pair-wise feature and is dependent on global scene features and the pair of nouns, and C^l is the context model which measures how well the l^{th} dimension of a pairwise feature R_{jk} satisfies the constraint learned for that dimension for the given pair of nouns.

Intuitively, equation 2.1 states that the cost can be minimized if: (1) We sum over the contextual constraints that have low cost, that is, Λ^t should only include informative edges. (2) the learned feature weights should be such that the dimensions which represent consistent relationships should have higher weight as compared to the other dimensions. Our goal is to minimize equation (1) with respect to all possible graphs in all training images and all possible weights. At that minima, we have a subset of edges for all the images in the training data-set and feature-weights at each edge. We then learn a non-parametric representation of $\mathcal{F}_e$ and β based on the importance and weights estimated for the edges in the training dataset. As we can see, the estimation of β in training images depends on edges that are important in the training images and the evaluation of the importance of edges depends on β. Therefore, we employ an iterative approach where we fix β and learn the function $\mathcal{F}_e$ and in the next step, based on the importance of edges in the training dataset, we re-estimate β.

2.4.1 Iterative Approach

Learning $\mathcal{F}_e$: Given feature-weights β, we predict whether an edge is informative or not. The information carried by an edge (representing potential contextual constraints on the pair-wise assignment of nouns to the nodes at the two ends of the edge) is a measure of how important that generic edge type is for inferring the labels associated with the nodes connected by the edge. The information carried in an edge depends on both global and local factors such as viewpoint and discriminability. In-

stead, of discovering all the factors and then learning a parametric-function; we use a non-parametric representation of the importance function. However, we still need to compute the importance of each edge in the training data-set.

To compute the importance of an edge, we use the message-passing algorithm. The importance of an edge is defined as how much the message passing through the edge helps in bringing the belief of nodes connected by the edge towards their goal belief (ground-truth). Suppose that the goal beliefs at node i and j are y_i and y_j respectively. The importance of the edge between i to j is defined as:

$$I(i \leftrightarrow j) = \frac{1}{iter} \sum_{k=1}^{iter} (y_i.b^k_{\mathcal{N}_i} - y_i.b^k_{\mathcal{N}_i - (i,j)}) + (y_j.b^k_{\mathcal{N}_j} - y_j.b^k_{\mathcal{N}_j - (i,j)}) \tag{2.3}$$

where $b^k_{\mathcal{N}_i}$ is the belief at node i at iteration k computed using messages from all the nodes (fully-connected setting); $b^k_{\mathcal{N}_i - (i,j)}$ is the belief at node i at iteration k computed using messages from all the nodes except $i \leftrightarrow j$ (edge-dropped setting). *iter* is the total number of iterations of message passing algorithm.

Using this approach, the importance of each edge is computed based on the local message passing algorithm. It does not take into account the behavior of other similar edges (similar global scene features and connecting similar local regions) in the training dataset. For example, in a particular image from the set of beach scenes, the edge between sky and water might not be important; however if it is important in most other beach images, we want to increase the importance of that particular edge so that it is not dropped. We therefore update the importance of an edge by using the importance of the edges which have similar global and local features.

This is followed by an edge dropping step, where the edges with low importance are dropped randomly to compute an efficient and accurate networks for the training images.

Learning β: Given the importance function of the edges $\mathcal{F}_e()$, we estimate β. As stated above, we use locally weighted regression for estimating β, therefore we need to estimate individual feature weights for all edges. Given the cost function in equation 2.1, a gradient descent approach is employed to estimate the feature weights of edges. We obtain the gradient as:

$$\frac{\partial \mathcal{C}}{\partial \beta^l_{n_j,n_k}} = C^l(G^t, R_{jk}) \tag{2.4}$$

where $\beta^l_{n_j,n_k}$ is the weight of l^{th} feature for edge (j,k). The above equation states that for a given pair of nouns, if the l^{th} dimension of pairwise feature is consistent with the l^{th} dimension of pairwise features from similar images, then the value of $\beta^l_{n_j,n_k}$ should be increased. Therefore, the value of β is updated at each step using the gradient above and then normalized ($\sum_l \beta^l = 1$). Intuitively, this equation evaluates which contextual relationship is satisfied on average for a pair of nouns and increases it weight. For example, between sky and water the above relationship is always satisfied where as left/right has high variance (In images sky is sometimes on left and sometimes on right of water). Therefore, this equation increases the weight of dY (measuring above) and decreases the weight of dX (measuring left).

2.4.2 Inference

Given a segmentation of an image, we construct a Markov network for the image using the function $\mathcal{F}_e$. For this construction, we first compute the global features, G, of the image and the local features of every region in the segmentation. A potential edge in the network is then predicted using simple locally weighted regression:

$$\mathcal{F}_e(G, R_j, R_k) = \sum_{t, j_t, k_t} W(G, G^t, R_j, R_{j_t}, R_k, R_{k_t}) M(j_t \leftrightarrow k_t) \tag{2.5}$$

where $W()$ is the weight function based on distances between local and global features of training and test data and $M()$ is an indicator function which predicts whether the edge was retained in the training data or not. The feature weights are also computed using locally weighted regression. The labels are then predicted using the message passing algorithm over the constructed Markov network with the estimated feature weights.

2.5 Experimental Results

We describe the experiments conducted on a subset of the LabelMe [25] dataset. We randomly selected 350 images from LabelMe and divided the set into 250 training and 100 test images. Our training data-set consists of images with segmentations and labels provided [1]. We used GIST features [26] as global features for scene

[1] grass, tree, field, building, rock, water, road, sky, person, car, sign, mountain, ground, sand, bison, snow, boat, airplane, sidewalk

matching. For appearance modeling, we use Hoeim's features [27] together with class specific metric learning used by [28]. The pairwise relation feature vocabulary consists of 5 contextual relationships [2]. In all experiments, we compare the performance of our approach to a fully-connected Markov network and a neighborhood based Markov network. We measure the performance of our annotation result as the number of pixels correctly labeled divided by total number of pixels in the image, averaged over all images.

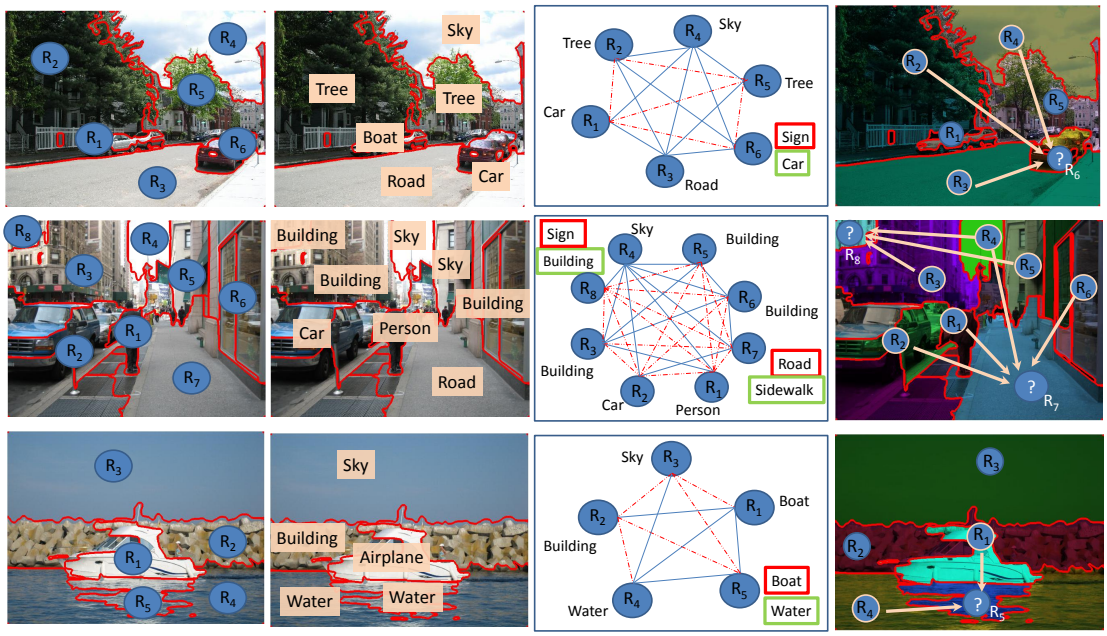


Figure 2.5: A few qualitative examples from the LabelMe dataset of how constructing the network structure using our approach leads to an efficient Markov structure which also improves labeling performance

In the training phase, we run inference on each training image and utilize the ground truth to evaluate the importance of each edge. At each iteration, a few

[2]Contextual relations - above/below, left/right, greener, bluer, brighter

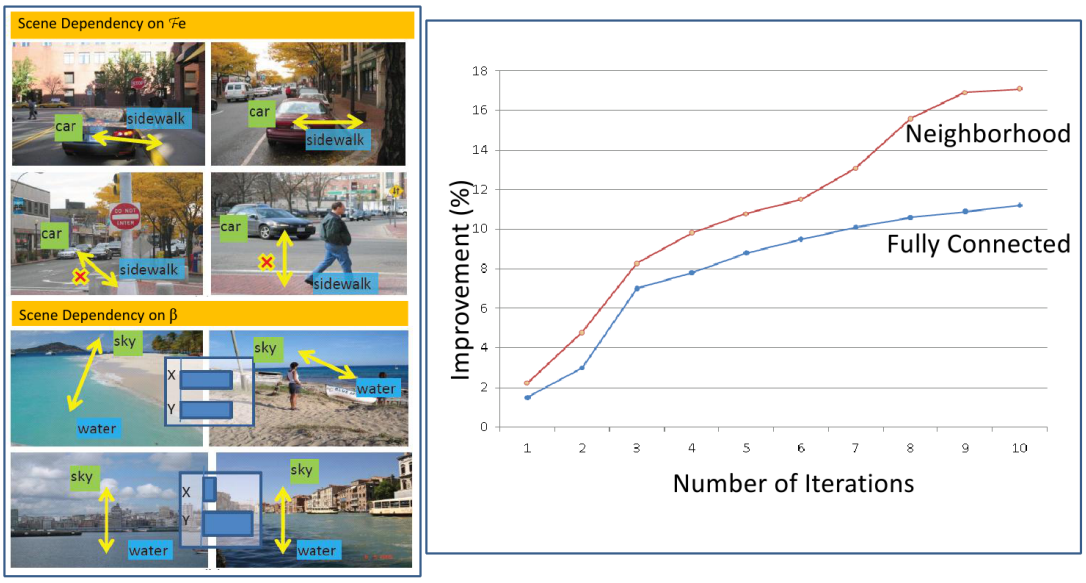


Figure 2.6: (a) Scene dependency of $\mathcal{F}(e)$ and β. In the first case the edge between sidewalk and road is informative only when the car is parked or is nearby it. In the second case, in scenes like beaches, both x and y are important features of context; however when the viewpoint is such that water occupies most of the lower space, the importance of x decreases. (b) The graphs show the % improvement over the fully-connected and neighborhood based Markov network as the training continues

unimportant edges are dropped and feature weights are re-estimated. Figure 2.6 (a) show examples of how our approach captures the scene dependency of $\mathcal{F}(e)$ and β respectively. Fig 2.6 (b) shows the percentage improvement over a fully-connected network and a neighborhood based network with each iteration. The figure clearly shows that dropping edges not only provides computational efficiency, but also improves the matching scores in training due to the removal of spurious constraints or constraints which link regions which are not discriminative.

On test images, we first predict the Markov network using the learned $\mathcal{F}(e)$

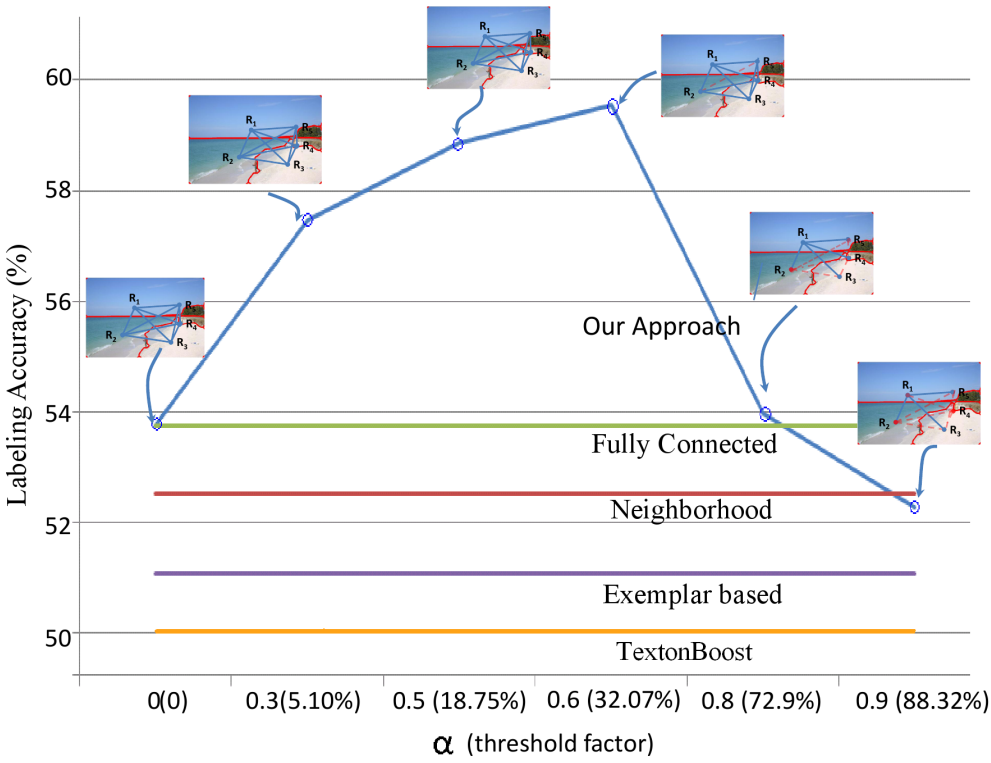


Figure 2.7: The graph shows the improvement of our algorithm over the fully-connected network and neighborhood based network on the LabelMe dataset with an example of graph structures at different thresholds of $\mathcal{F}(e)$. The values in the parentheses shows the percentage of edges dropped at a given threshold of $\mathcal{F}(e)$

and then utilize β to perform inference. Figure 2.7 show the performance of our approach compared to a fully-connected and a neighborhood connected Markov network on the LabelMe dataset at different thresholds of $\mathcal{F}(e)$. A higher threshold corresponds to dropping more edges. The values in parenthesis on the threshold axis shows the average percentage of edges dropped at that particular threshold. We also compared the performance of our approach to publicly available version of texton-boost (without CRF) on our LabelMe dataset and it yields approximately 50% as

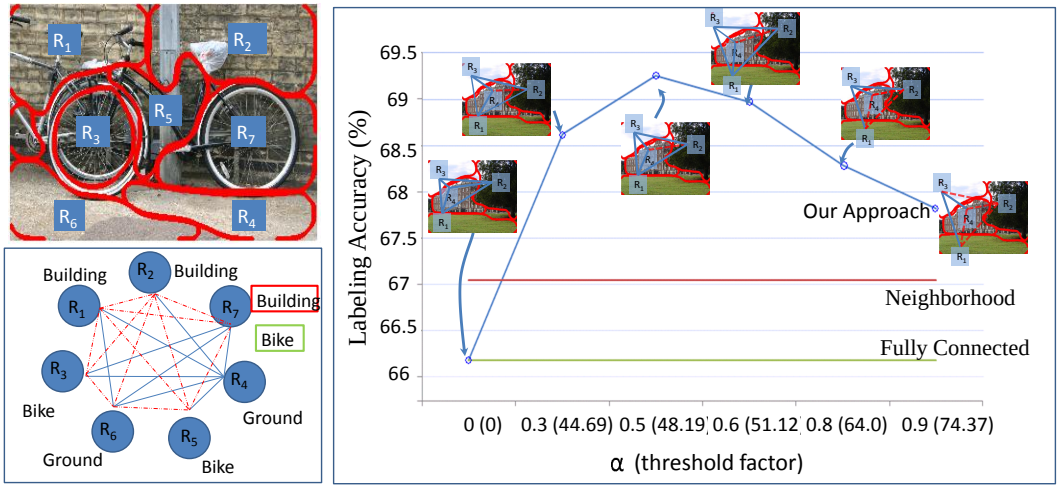


Figure 2.8: (a) An example where our approach removed spurious edges and improved labeling performance. (b) Labeling accuracy of our algorithm compared to fully-connected and neighborhood connected Markov networks on the MSRC dataset with examples of graph structures at different thresholds of $\mathcal{F}(e)$

compared to 59% by our approach. It should be noted that this is similar to the performance of the local-appearance based model used in our approach. Therefore, our approach should also provide considerable improvement over the CRF version of the texton-boost as well. Above the performance chart, we show "what" edges are dropped at a given threshold. We also compare our approach to the exemplar based approach similar to [29] where the labels are transferred based on the edge matches in the training dataset.

Figure 2.5 shows representative examples where our approach performed better than the fully-connected network. The second column of the figure shows the result obtained using just the appearance model (likelihood term). The third column shows our network compared to a fully-connected Markov network. The label marked in

red is the result obtained using the fully-connected network while the label in green is the result obtained using our approach. In the last column, we show the regions of interest (where our approach dropped spurious edges which led to improvement in performance). In the first example, if a fully-connected network is utilized, the label of the red car on the right side of road is forced to signboard (by other car). This is because the car-signboard relationship is stronger than car-car relation in the current spatial configuration and the bad appearance model predicts signboard as a more likely label. On the other hand, when the spurious edge is dropped, the labeling improves and the region is correctly labeled as a car. Similarly in the second example, we observe that the sidewalk is labeled as road in the fully-connected network (due to strong appearance likelihood and presence of buildings). On the other hand, the region labeled as person boosts the presence of sidewalk (people walk on sidewalks) and when spurious edges from buildings are dropped by our approach the labeling improves and the region is correctly labeled as sidewalk.

We additionally tested our algorithm on the MSRC dataset. The training and testing data of the MSRC dataset is the same as in [7]. The dataset has 21 object classes. It should be noted that MSRC is not an ideal dataset since the number of regions per image is very low and therefore there are not many spurious edges that can be dropped. However, this experiment is performed in order to compare the performance of our baseline to other state-of-the-art approaches. Figure 2.8(a) shows the performance of our algorithm compared to fully-connected and neighborhood connected networks on the MSRC dataset. Our results are comparable to the state of the art approaches based on image segmentation such as [7]. Figure 2.8(b) shows

an example of one case where dubious information is passed along edges in the fully-connected network leading to wrong labeling. Region 7, in the fully connected network, was labeled as building. The is because the building-building and bike-building contextual relationship is stronger than bike-bike relationship. But when the link of the bike region with regions labeled as building was removed through our edge prediction, it was correctly labeled as bike.

Chapter 3: Context in a Multiple Segmentation and Recognition Framework for Scene Labeling

3.1 Introduction

We describe an approach that jointly segments and labels the principal objects in an image. Consider the image in figure 1. Our goal is to locate and pixel-wise label the principal objects such as car, building, road and sidewalk. One approach is to first segment the image, then perform recognition using appearance and context. However, there are generally no reliable algorithms for segmentation. For example, for the image shown in Figure 3.1, segmentation algorithms will generally not combine the roof and the body of the car into one segment due to differences in appearances. Therefore, there has been a recent trend to simultaneously address segmentation and recognition.

For example, some recent approaches construct the segments by selectively merging superpixels while simultaneously labeling these elements. However, at the superpixel level global image features such as shape cannot be easily employed. So, while these approaches show high performance for "stuff"-like objects such as grass - they often fail to identify objects which require shape cues for identification.

To harness shape features, approaches such as [30, 31] have instead started with an initial segmentation and then refined these segments iteratively. However, the modifications are generally local in nature and tend to get stuck in local minima.

To overcome these problems, recent approaches have advocated the use of multiple segmentations [11,32]. Recognition, then, involves selecting the best segments. These methods use only appearance features to select segments and the best overall labeling is constructed in a greedy manner. They ignore context, which is important for accurate segment selection and labeling. For example, the window of the car is labeled as "airplane" because the context from other scene elements such as road, sidewalk and building are ignored.

We propose an approach to select the best segmentation and labeling in a single optimization procedure that utilizes context to perform segment selection and labeling coherently. To overcome the fragmentation problem, we allow connected segments to be merged based on local color, texture and edge properties. We also include mid-level cues to constrain the solution space - for example, the segment merging step leads to overlapping segments, and we restrict global solutions to exclude overlapping segments (avoiding the possibility of multiple labeling for pixels). By incorporating contextual relations between region pairs, we find the subset of segments that best explains the image. For example, in Figure 3.1, our approach correctly selects the combined region of window and body segments and labels it as "car". The labeling of the window segment as "airplane" is not chosen due to contextual constraints from sidewalk, road and building.

The contributions of our work are: (a) An approach to incorporate contextual

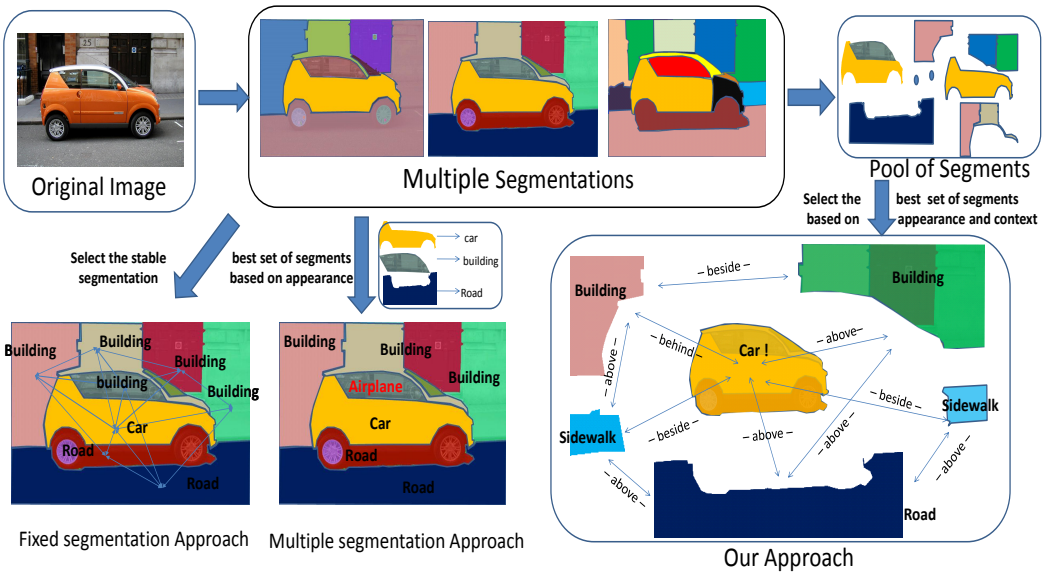


Figure 3.1: Comparison of our approach to fixed and multiple segmentation algorithms. Our approach solves the problem of segmentation and recognition jointly using appearance and context. The figure shows how global contextual relations help to select the whole car segment subset over other fragmented pieces of car, as their association does not satisfy context.

information in a multiple segmentation framework, and (b) Increasing the spatial support[1] of image labeling by constructing additional segments from a base pool, at the cost of only a small increase in segment pool size.

3.2 Related Work

The problem of image parsing has a long history in computer vision dating back to the 1970's. Unlike Marr's sequential processing pipeline, where segmenta-

[1]Spatial support measures the quality of pool of segments as compared to ground truth. The score is higher if the segments in the ground-truth find segments in the pool with high overlap.

tion from bottom-up cues preceded recognition, Tenenbaum and Barrow proposed Interpretation-Guided Segmentation [33] which labeled image regions using constraint propagation to arrive at a globally consistent scene interpretation. This was followed by development of complete scene understanding systems such as ACRONYM [34] and VISONS [35]. During the last decade, researchers in visual recognition have made significant advances in object recognition due to better appearance modeling techniques and visual context. These approaches can be broadly categorized into three categories based on how interactions between segmentation and recognition are modeled:

Pixel Based Approaches: These approaches model the problem of visual recognition at the pixel level [7, 36–38] and therefore the problem of segmentation is solved implicitly (neighboring pixels belonging to different class represent boundary pixels). One of the major shortcomings of pixel-based approaches is that many objects (such as cars) are defined in large part by their shape and therefore categorization at the pixel-level using local appearances without global shape analysis performs poorly.

Fixed Segmentation Approaches: These approaches classify individual regions in some fixed image segmentation based on region color, texture and shape [3, 4,39]. However, obtaining semantically meaningful segmentations without top-down control is well beyond the state of the art.

Image Parsing (Joint Segmentation and Recognition): These approaches jointly solve segmentation and recognition. Approaches such as [10, 11] obtain multiple segmentations of the image and model the problem of segmentation and recog-

nition as the selection of segments based on their matches to semantic classes. On the other hand, approaches such as [30,31,40] start from an imperfect segmentation and then refine it iteratively by optimizing a cost function defined on segments and appearance matching. One of the shortcomings of these approaches is that they tend to get stuck in local minima due to local refinement. [41, 42] proposed super pixel based approaches where the class labels are inferred based on local appearance and context using CRFs. Such approaches fail to incorporate higher level shape information; additionally learning CRF's parameters has proven to be difficult. In [19] segmentation was combined with the responses of sliding window object detectors for image labeling to avoid fragility of segmentation.

3.3 Overview

Multiple segmentation approaches construct a pool of initial segments by varying the controlling parameters of a segmentation algorithm or by starting from a coarse segmentation and iteratively refining the segmentation by merging or further segmenting initial segments. They generally assume that each object will be well segmented at some parameter setting or level. [43] pointed out that merging small connected subsets (pairs and triples) of base segments improves recognition performance. However, the algorithm in [43] employed manually choosing the segments to merge. One could simply join all possible pairs and triples of connected segments but this would lead to an explosion in the segment pool size. In contrast, we construct a "good" set of mergings using a classifier which rejects combination which

are unlikely to correspond to "complete" objects (section 4).

We organize these segments into a hierarchical segment graph for recognition. The graph structure allows us to impose constraints that reduce the combinatorics of the search process - for example, that a solution cannot include overlapping segments, since this could lead to pixels being given multiple labels.

Given the segment graph, we compute pairwise and higher-order constraints on selection of segments. We then formulate a cost function which accounts for local appearance and enforces pair-wise contextual relationship consistency (such as sky above water, road below car, etc). Directly optimizing this cost function is NP hard so the cost function is approximately minimized by first relaxing the selection problem. The relaxed problem can be solved efficiently by quadratic programming (QP). The relaxed solution is then discretized to obtain the final labeled segmentation (section 5). Finally, we evaluate the performance of our approach with previously reported methods (section 6).

3.4 Constructing the Segment Graph

Obtaining the Initial Segment Pool: We use the hierarchical segmentation algorithm from [44] to construct the segment pool. To increase the robustness of the segmentation algorithm, we use the stability based clustering analysis of [16]. Stability analysis selects segments which are stable under small perturbations (noise) to the image.

In the first step, image is segmented and the segments in the first hierarchical

level are added to the segment pool. Then each of these segments is iteratively segmented and the smaller segments are added to the segment pool until any of the following conditions are met. (1) The segment size is too small ($< 2\%$ of total image pixels). (2) The integrated edge strength along the boundary of the segment (obtained by Berkeley edge detector [45]) is below a threshold. (3) The number of leaf nodes in the segment subgraph rooted at the original segment exceeds a threshold.

This procedure gives us initial segment pool over which we will perform segment selection.

Merging Segments: The base segmentation algorithm seldom produces segments that directly correspond to the objects in the image. Hence, we merge small (2 and 3) connected sets of segments from the segment pool to obtain a better collection of segments. But allowing all possible segment merges would explode the size of the pool. To limit the number of pairs and triples merged, we learn a function that scores these small subsets from a training set of fully labeled images.

A Support Vector Regression (SVR) [46] model using radial basis functions is learned from the training images to score potential merges. We compute color, texture and edge features similar to those used by Hoiem et. al. [47] for each segment of an object. Based on these features, the SVR predicts whether the segments should be merged or not. Training images are segmented using the segmentation algorithm described above and a segment pool is obtained for each image. Objects which are broken into multiple segments are determined using the ground truth segmentation. These fragmented objects provide positive examples and the negative examples are

obtained using random samplings from the training data. For a testing image, each adjacent pair and connected triple[2] of segments is evaluated for merging using the regression model learned, providing a score for each merging. The pairs and triples with scores above a threshold are added to the segment pool.

We evaluated the merging scheme on the 256 test images in the MSRC dataset. Figure 3.2 shows the spatial support in the pool with increasing pool size. The pool size is increased by lowering the threshold at which mergings are accepted. To demonstrate that the SVR learns an informative merging function, we compare the spatial support metric when the segment pool is enlarged using random merges (red curve in Figure 2). Although spatial support increases (which it obviously must), it does so at a much slower rate than the SVR.

Construction of the Segment Graph: The pool of segments are then arranged in a hierarchical graph structure to which our inference algorithm will subsequently be applied. The graph structure is constructed as follows: The root node is assigned to the whole image. A segment S_i is a child of segment S_j if segment $S_i \subset S_j$. If two segments S_i and S_j are subsets of a S_k then both the segments are children of segment S_k. The segments which have no smaller segment subsets are leaf nodes.

[2]triples of segments are constructed by evaluating merging of a segment from the initial pool with an adjacent segment formed from the pairwise merging step.

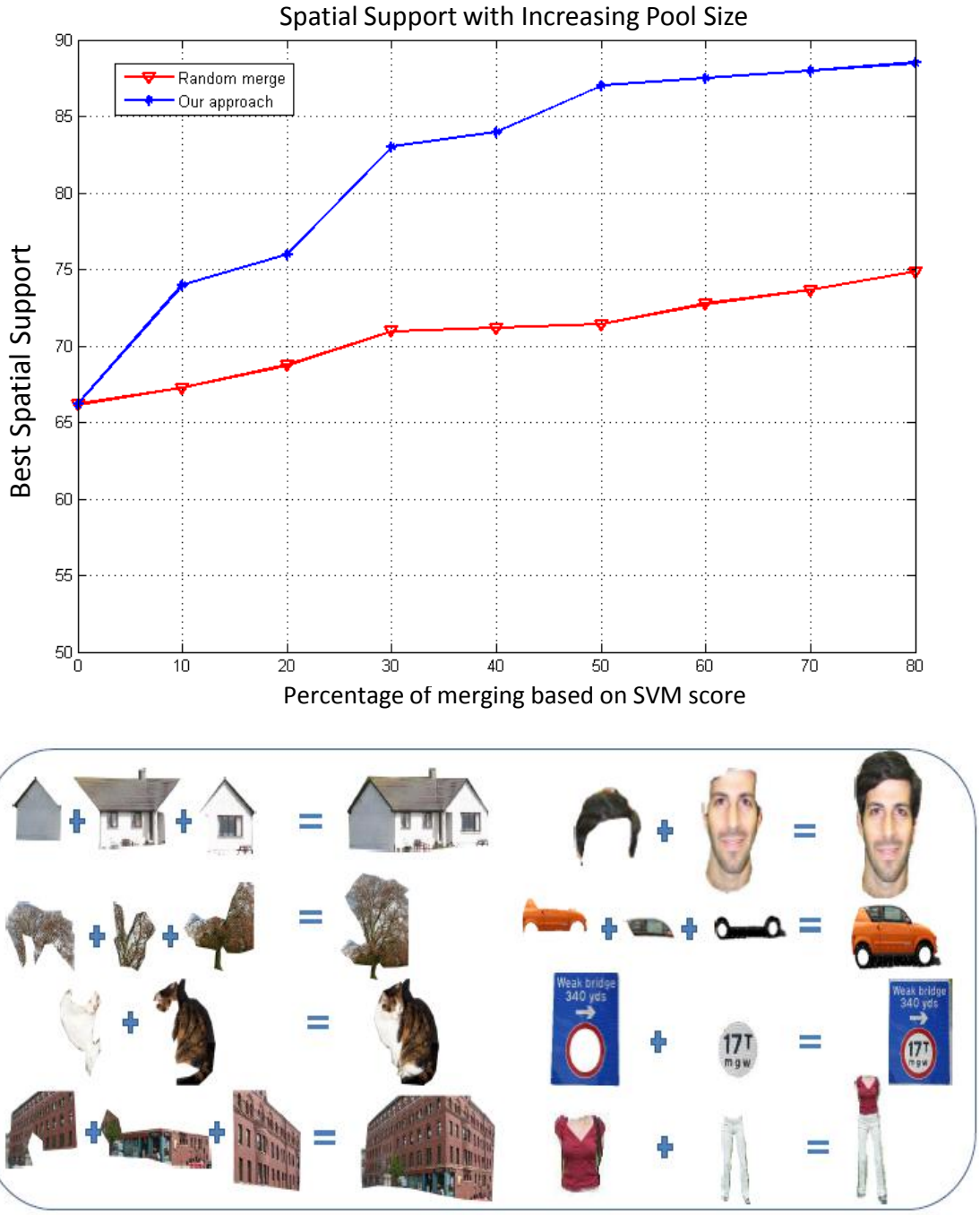


Figure 3.2: Graph on top shows the improvement in spatial support with increase in pool size. Image below the graph shows the instances where SVR model correctly merged fragmented segments of objects in the pool to complete the object segment.

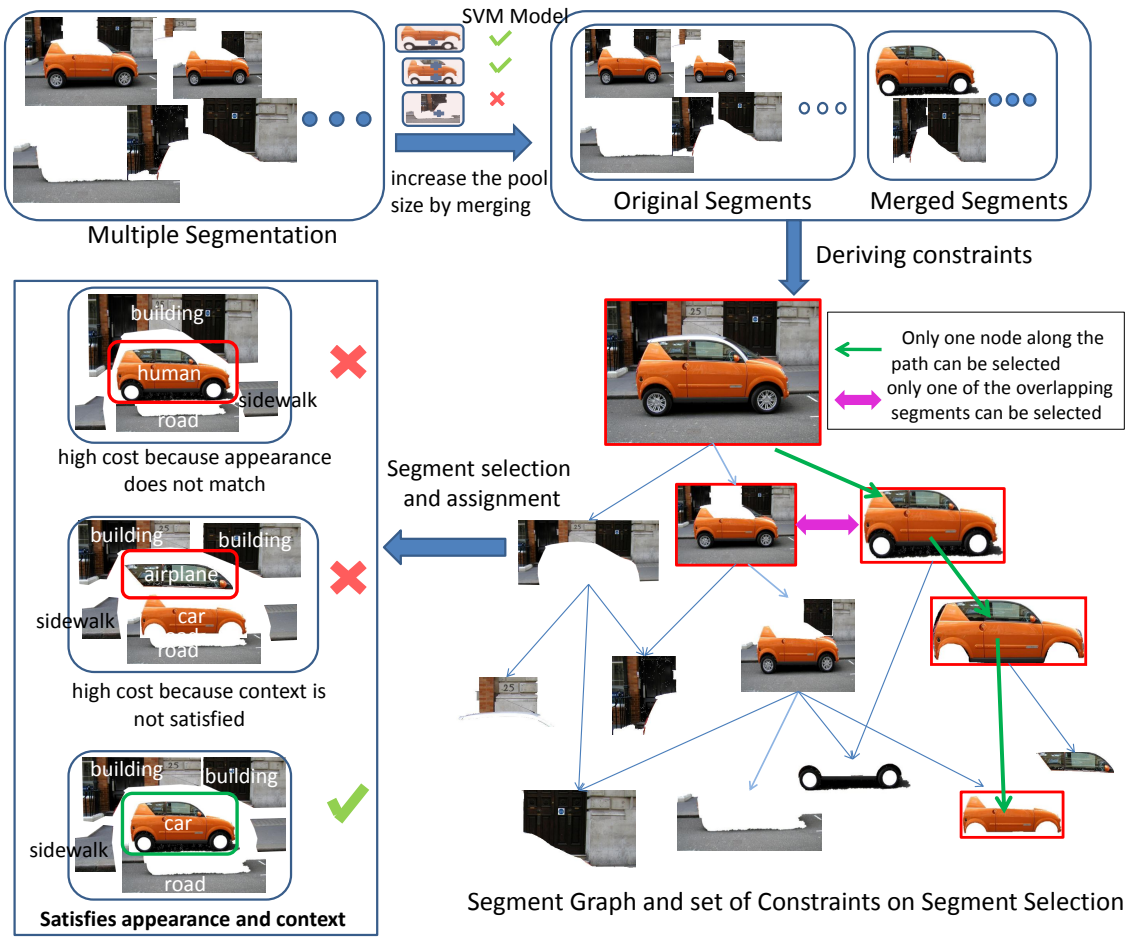


Figure 3.3: Our approach: We first create a pool of segments using multiple segmentations of an image and merging some of the connected pairs and triples of these segments. These segments are arranged in a graph structure where path constraints are used to obtain selection constraints. An example of a path constraint is shown using green edges: only one segment amongst all the segments in the path can be selected. The magenta arrow shows that two segments which overlap cannot be selected simultaneously. Finally, the QP framework is used to find the set of segments, together with their labels, which minimizes the cost function given the constraints

3.5 Piecing together the Segments

Our goal is to select a set of segments from the pool such that each segment has high overlap with a ground-truth segment and is assigned its correct label.

We formulate a cost function which evaluates any possible selection and labeling of segments from the pool. Each segment, S_i in the pool is associated with a binary variable X^i which represents whether or not the segment is selected. With each selected segment we also associate a set of C binary variables, $(X^i_1 ... X^i_C)$, which indicates the label associated with the segment. $X^i_j = 1$ represents that segment i is labeled with class j. Our goal is to choose X^i such that the cost-function $\mathcal{J}$ is minimized, where $\mathcal{J}$ is defined as:

$$\mathcal{J} = \sum_{i,j} -w_1 A_{ij} X^i_j - \sum_i w_2 S_i X^i + \sum_{i,j} \sum_{k,l} w_3 X^i_j P_{ijkl} X^k_l \tag{3.1}$$

The cost function consist of three terms. The first term uses an appearance based classifier to match the appearance of selected segments with their assigned labels. The second term is the explanation reward term which rewards the selection of segments proportional to their size. The third term is a context satisfaction term which penalizes assignments which do not satisfy the contextual relationships learned from the training data. We discuss each of these terms below. The weight w_1,w_2,w_3 are obtained by cross validation on a small dataset and for our experiments we use 1, 1.5 and 0.5 respectively.

3.5.1 Constraints on Segment Selection

While there are 2^{N_S} possible selections (where N_S is the number of segments in the pool), not all subsets represent valid selections. For example, if segment i is selected and assigned label j, then other segments which overlap with segment i should not be selected to avoid multiple labeling of pixels. Figure 3.3 shows the overlap constraint by a magenta arrow where the two car segments which overlap cannot be chosen simultaneously. Similarly, two segments along a path from the root to any leaf node cannot be selected together. Figure 3.3 shows one such path constraint in green, where selection of the car and its subset segments simultaneously is prohibited.

These constraints are represented as follows:

$$0 \leq X^i + X^k \leq 1 \ \ \forall (i,k) \in O \tag{3.2}$$

$$0 \leq X^{p_1} + X^{p_2} .... X^{p_m} \leq 1 \ \ \forall p \in \mathcal{P} \tag{3.3}$$

where O represents the set of pairs of regions in the graph that overlap spatially and $\mathcal{P}$ represents the set of paths from the root to the leaves in the segment graph. Additional constraints that are enforced while minimizing the cost function $\mathcal{J}$ include:

$$0 \leq X^i \leq 1 \tag{3.4}$$

$$\sum_j X^i_j = X^i \tag{3.5}$$

These constraints allow only one label to be assigned to each selected segment.

3.5.2 Cost Function

We now explain the individual terms in the cost function.

Appearance Cost: The first term in the cost function evaluates how well the appearance of the selected segment i associated with label j matches the appearance model for class j. For computing A_{ij}, we learn an appearance model from training images using a discriminative classifier over visual features. We use the appearance features from [47] and learn a discriminative probabilistic-KNN model as in [28, 48] for classification.

Explanation Reward: This term rewards selecting a segment proportional to its size, represented by S_i. This term avoids the trivial solution where no segment gets selected by the algorithm.

Contextual Cost: The third term evaluates the satisfaction of contextual relationships for a given selection of segments and their label assignment. We model context by pair-wise spatial and contextual relationships as in [3]. If segment i is assigned to class j and segment k is assigned to class l, P_{ijkl} measures the contextual compatibility based on co-occurrence statistics of classes j and l. We also evaluate spatial contextual compatibility by extracting the pairwise-differential features as in [3] for segments i and k and comparing them with a learned model of differential features for labels (j, l). For example, if the labeling is such that sky occurs below water then the penalty term is kept high and vice-versa. The penalty term is defined as:

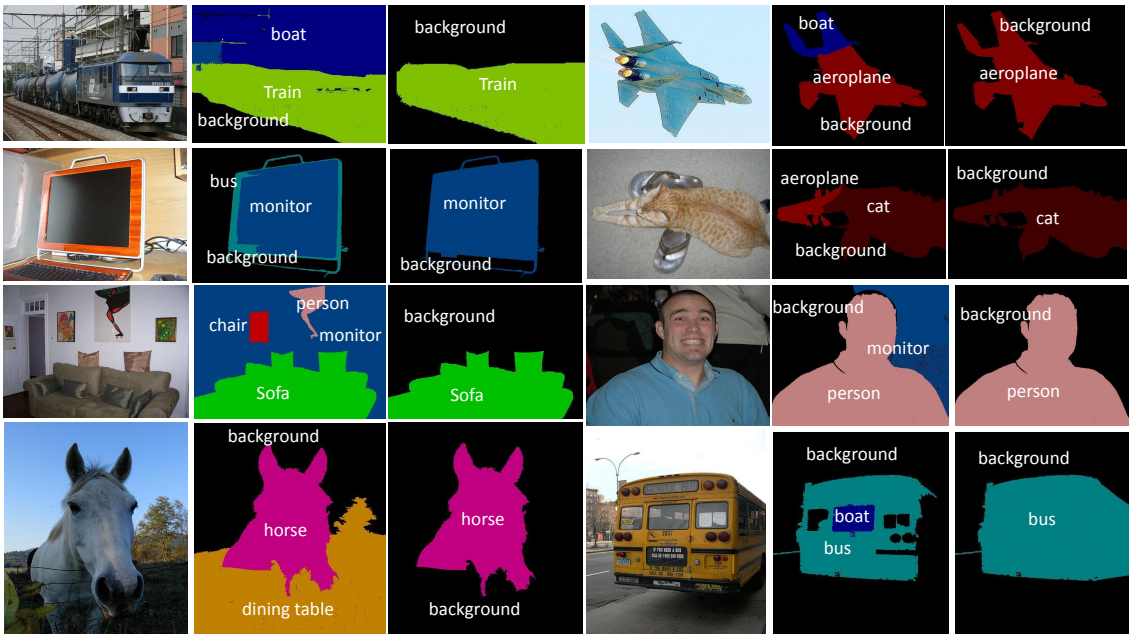


Figure 3.4: PASCAL VOC'09 labeling results. Columns (a) and (d) - original images. Columns (b) and (e) show the performance of appearance based approach without context. Columns (c) and (f) show the performance of our algorithm with context. Best viewed in color.

$$P_{ijkl} = C_1 \exp(\frac{(d_{i,k} - \mu_{j,l})^2}{2\sigma_{j,l}^2}) + C_2 \exp(-\alpha M_{j,l}) \tag{3.6}$$

where C_1, C_2 and α are constants. $d_{i,k}$ is the differential feature between segment i and segment k. $\mu_{j,l}$ is the mean differential feature obtained from training between class labels j and l. The term $M_{j,l}$ represents the co-occurrence of classes j and l, also obtained from training. We employ eight differential features - Δx,Δy,$\Delta \mu_{red}$,$\Delta \mu_{green}$,$\Delta \mu_{blue}$,$\Delta \mu_{brighter}$, adjacency and overlap.

Table 3.1: Performance comparison of our algorithm against previous approaches on PASCAL VOC09 dataset.

	Background	Airplane	Bicycle	Bird	Boat	Bottle	Bus	Car	Cat	Chair	Cow	Dining Table	Dog	Horse	Motor Bike	Person	Potted Plant	Sheep	Sofa	Train	TV/Monitor	Average
Hierarchical CRF [42]	77.7	38.3	9.6	**24.0**	35.8	**31.0**	59.2	36.5	21.2	8.3	1.7	22.7	14.3	17.0	26.7	21.1	15.5	16.3	14.6	48.5	33.1	27.3
Hierarchical CRF with CO [42]	82.3	**49.3**	**11.8**	19.3	37.7	30.8	**63.2**	46.0	23.7	10.0	0.5	23.1	14.1	22.4	33.9	35.7	**18.4**	12.1	**22.5**	**53.1**	37.5	30.8
Ours (w/oContext,w/Merging)	76.4	25.6	8.0	14.2	**47.3**	8.1	30.5	53.7	50.1	**18.6**	9.1	48.5	10.9	15.8	33.8	47.3	10.2	15.7	11.2	48.6	35.2	29.5
Ours (w/Context, w/oMerging)	61.2	37.3	5.5	20.6	36.0	14.6	30.8	55.3	46.8	10.6	4.2	40.2	11.3	17.3	29.0	36.1	9.1	29.3	12.8	47.4	38.2	28.3
Ours (Context,w/Merging)	**85.8**	39.8	7.6	18.4	45.0	8.4	44.6	**66.1**	**54.2**	11.2	**10.3**	**52.7**	**15.2**	**23.5**	**39.2**	**50.8**	11.5	**31.5**	19.8	40.4	**48.9**	**34.5**

3.5.3 Optimization

For optimizing the cost function, we relax the binary variables X^i and X^i_j to lie in [0, 1]. We use the Integer Projected Fixed Point (IPFP) algorithm [49] to minimize the cost function.

The solution generally converges in 5-10 steps, which makes it very efficient, while outperforming current state-of-the-art methods for inference. IPFP solves quadratic optimization functions of the form:

$$x'^* = argmax(x'^T M x') \;\; s.t. \, Ax' = 1, \; x' \geq 0 \tag{3.7}$$

To use the IPFP algorithm, we transform the original equation 1 into 7 through the following substitution: $x' = \left(\begin{smallmatrix} 1 \\ X \end{smallmatrix}\right)$ and $M = \left(\begin{smallmatrix} 0 & (A+S)^T/2 \\ (A+S)/2 & -P \end{smallmatrix}\right)$. The path constraints discussed in section 5.1 are incorporated as constraints in a linear solver during step 2 of the optimization algorithm.

In the second step, the relaxed solution is then discretized to obtain an approximate solution. Here, higher probability segments are selected first and assigned their class labels as long as segment selection constraints are satisfied.

3.6 Experiments

We evaluated the performance of our algorithm on three standard dataset: Label Me subset (used in [39]), PASCAL VOC 2009 [50] and MSRC [37].

LABEL-ME: [39] used a subset of LABEL ME containing 350 images - 250 training and 100 testing. The dataset contains 19 classes. Performance is measured

Table 3.2: Performance comparison of our algorithm against other approaches on LabelMe dataset.

	Texton-boost	**MRF based**	**Jain et.al. [39]**	**Ours(no Context,Merging)**	**Ours(Context,no Merging)**	**Ours(Context, Merging)**
pixel wise	49.75	54.2	59.0	65.23	71.9	75.6
class wise	20	30.2	–	38.5	43.5	45

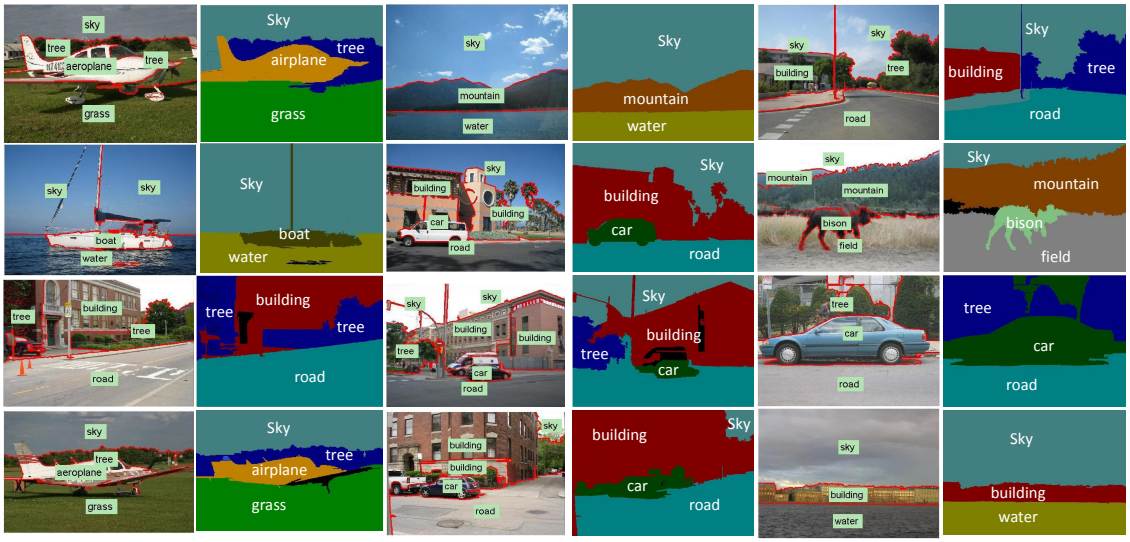


Figure 3.5: LabelMe dataset results - columns 1, 3 and 5 show the original image with object labels obtained by our algorithm and columns 2, 4 and 6 show the corresponding image segmentation.

using the two standard measures from [39]. For comparison, we also evaluate four approaches in addition to those compared in [39] (1) Our multiple segmentation framework, but without contextual information. (2) A fully connected MRF-model similar to [4], which performs recognition using context on a fixed segmentation obtained using stability analysis. (3) A Texton-boost approach [3] without the CRF model, and 4) our method applied to the initial segment pool, but without the SVR merged segments.

Figure 3.5 shows a few qualitative examples of our approach. When context is not utilized many small segments are mislabeled and matched to wrong object classes. However, when context is added many of these errors are eliminated.

Table 3.2 shows the quantitative performance of our approach compared with these four methods and [39] using the two standard evaluation metrics. Our approach has a pixel-wise accuracy of 75.6%; when only appearance is used the performance falls to 65.23%. This shows that contextual information is critical not only for recognition but also for segment selection. As expected, the fixed segmentation MRF model has a low pixel-wise accuracy of 54.2%. The publicly available version of Texton-boost achieves just 49% pixel-wise accuracy. This is because Texton-boost relies on pixel-based appearance models. These are adequate for modeling regions like 'grass' and 'sky' but perform poorly for objects whose recognition requires cues such as shape.

PASCAL VOC 2009: The PASCAL VOC 2009 dataset [50] consists of 1499 images which is split into 749 images for training and 750 images for validation. We

[3] http://jamie.shotton.org/work/code/

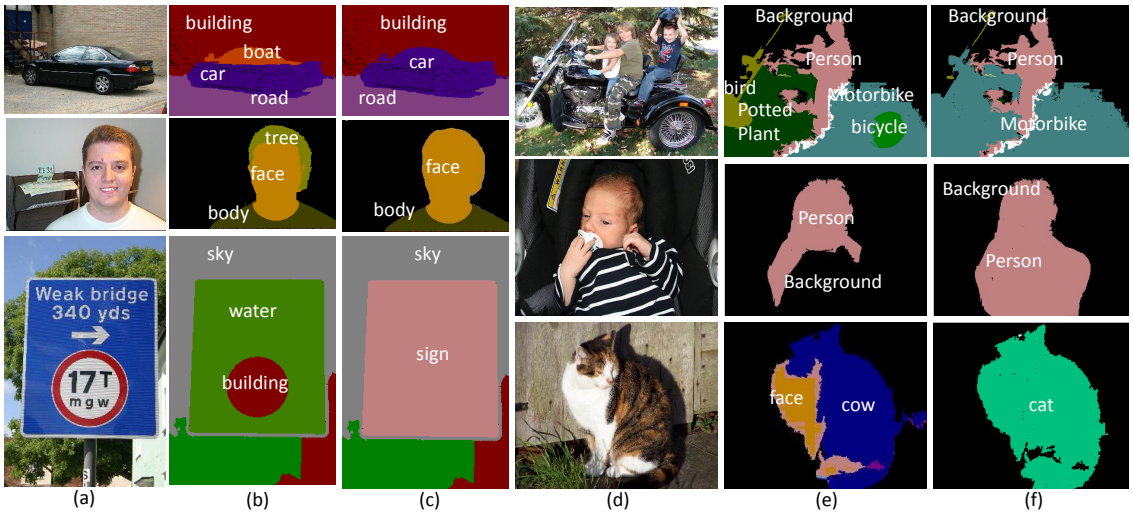


Figure 3.6: Qualitative results of our algorithm with and without merging. Columns (a) and (d) are original images. Columns (b) and (e) show the labeling performance without merging. Columns (c) and (f) show performance with merging. Best viewed in color.

follow the protocol used by [42] to compare against the state of the art, and use the same evaluation metric as [42]. Table 3.1 shows the class wise performance of our approach compared with the other approaches. Our approach outperforms previous approaches on many classes which shows that it generalizes to a large number of object classes. Our better performance on classes like Car, Cat, Horse, Sheep, Cow, Monitor, Dog and Person supports our contention that a multiple segmentation approach performs better on object classes for which shape is important. Table 3.1 also shows that both context and merging improves recognition by choosing segments which have better spatial support.

Figure 3.4 shows some qualitative results on VOC 2009. Columns (b) and (e)

show the labeling performance of our algorithm solely based on appearance. The algorithm using only appearance leads to a variety of errors such as the wing of the airplane being labeled as boat, the ground in the horse image as dining table, and the painting above the sofa as a person. Columns (c) and (f) show the performance of our approach with context. Figure 3.6 compares qualitative results of our algorithm with and without mergings and elucidates the importance of merging for better recognition. For example, in the sign image, the parts of the sign board are labeled as water and building but after merging them, it is correctly labeled as sign board.

MSRC dataset: Our algorithm achieved 75% (pixel-wise) and 68.7%(class-wise) on the MSRC dataset, which is comparable to state-of-the-art results except [42]. MSRC is relatively simple and does not significantly benefit from the use of multiple segmentations. Our approach performs better than [42] for classes like bird, car and cow, where multiple segmentation and merging helps by creating segments whose shapes are closer to class models, but performs poorer on "stuff" classes such as grass and sky.

Chapter 4: Representing Videos using Mid-level Discriminative Patches

4.1 Introduction

Consider the video visualized as a spatio-temporal volume in Figure 4.1a. What does it mean to understand this video and how might we achieve such an understanding? Currently, the most common answer to this question involves recognizing the particular event or action that occurs in the video. For the video shown in the figure it would simply be "clean and jerk" (Figure 4.1b). But this level of description does not address issues such as the temporal extent of the action [51]. It typically uses only a global feature-based representation to predict the class of action. We additionally would like to determine structural properties of the video such as the time instant when the person picks up the weight or where the weights are located.

We want to understand actions at a finer level, both spatially and temporally. Instead of representing videos globally by a single feature vector, we need to decompose them into their relevant "bits and pieces". This could be addressed by modeling videos in terms of their constituent semantic actions and objects [52–54]. The general framework would be to first probabilistically detect objects (e.g, weights, poles, people) and primitive actions (e.g, bending and lifting). These probabilistic

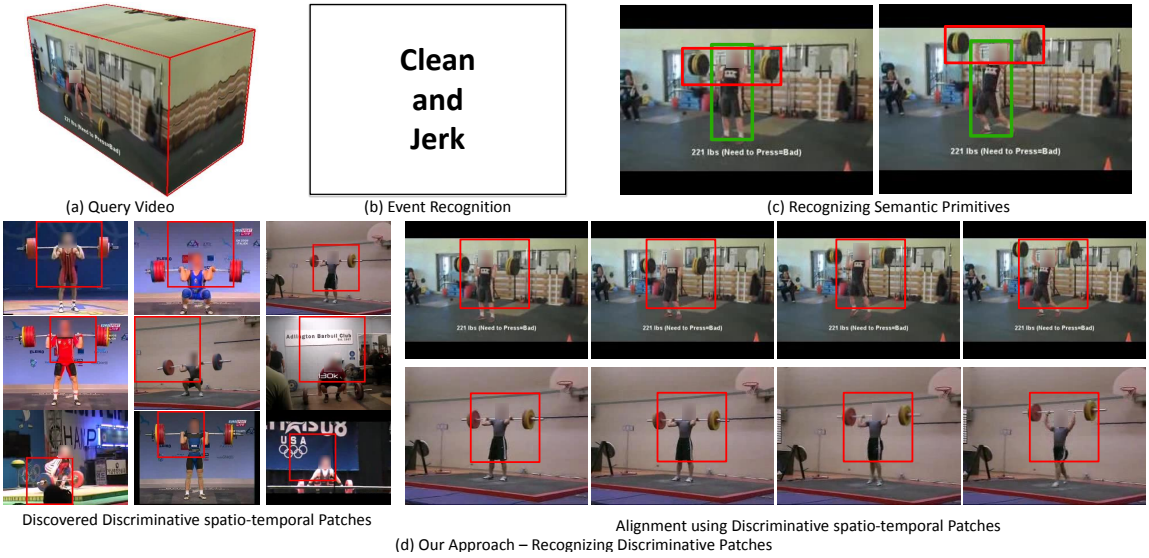


Figure 4.1: Given a query video (a), one can represent it using global feature vector and use it for action classification (b). Another possible representation is to use constituent semantic entities (c) and use object/action detectors for understanding. Instead, we propose a mid-level representation for videos (d). Our approach discovers representative and discriminative spatio-temporal patches for a given action class (d-left). These patches are then used to establishing correspondence followed by alignment (d-right). Additional examples are shown in the supplementary material.

detections could then be combined using Bayesian networks to build a consistent and coherent interpretation such as a storyline [55] (Figure 4.1c). So, the semantic objects and actions form primitives for representation of videos. However, recent research in object and action recognition has shown that current computational models for identifying semantic entities are not robust enough to serve as a basis for video analysis [56]. Therefore, such approaches have, for the most part, only been applied to restricted and structured domains such as baseball [55] and office scenes [53].

Following recent work on discriminative patch-based representation [57, 58], we represent videos in terms of discriminative spatio-temporal patches rather than global feature vectors or a set of semantic entities. These spatio-temporal patches might correspond to a primitive human action, a semantic object, human-object pair or perhaps a random but informative spatio-temporal patch in the video. They are determined by their discriminative properties and their ability to establish correspondences with videos from similar classes. We automatically mine these discriminative patches from training data consisting of hundreds of videos. Figure 4.1(d)(left) shows some of the mined discriminative patches for the "weightlifting" class. We show how these mined patches can act as a discriminative vocabulary for action classification and demonstrate state-of-the-art performance on the Olympics Sports dataset [59] and the UCF-50 dataset[1]. But, more importantly, we demonstrate how these patches can be used to establish strong correspondence between spatio-temporal patches in training and test videos. We can use this corre-

[1]http://server.cs.ucf.edu/vision/public_html/data.html

spondence to align the videos and perform tasks such as object localization, finer-level action detection etc. using label transfer techniques [60, 61]. Specifically, we present an integer-programming framework for selecting the set of mutually-consistent correspondences that best explains the classification of a video from a particular category. We then use these correspondences for representing the structure of a test video. Figure 4.2 shows an example of how aligned videos (shown in Figure 4.1(d)(right)) are used to localize humans and objects, detect finer action categories and estimate human poses.

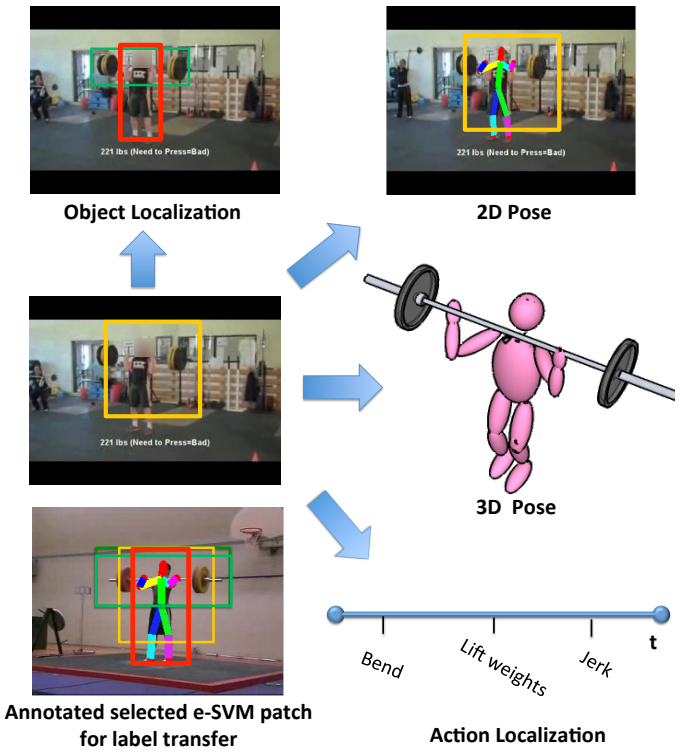


Figure 4.2: Strong alignment allows us to richly annotate test videos using a simple label transfer technique.

4.2 Prior Work

Prior approaches to video representation can be roughly divided into three broad categories. The first and earliest represent actions using global spatio-temporal templates, such as motion history [62] and spatiotemporal shapes [63] .

The second class of approaches is based on bag of features models [64–66], where sparse spatio-temporal interest points, dense interest points [67], page-rank features [68], or discriminative class-specific features [69], are computed as part of a bag of words representation on local features. Typically these representations are most appropriate for classification; they are not well-suited as action detectors or for establishing correspondence.

The third class of approaches is structural and decomposes videos into constituent parts. These parts typically correspond to semantic entities such as humans and objects [52, 53, 70]. While these approaches attempt to develop a rich representation and learn the structure of the videos in terms of constituent objects, one of their inherent drawbacks is that they are highly dependent on the success of object and action detection algorithms. Therefore, such approaches have not been used for "data in the wild". A more recent approach is based on using discriminative spatio-temporal patches rather than semantic entities [1, 71]. For example, [1] uses manually selected spatio-temporal patches to create a dictionary of discriminative patches for each action class. These patches are then correlated with test video patches and a new feature vector is created using pooling. There are several issues here: 1) What is the criteria for selecting spatio-temporal patches to create the dic-

tionary? 2) How many patches are needed to capture all the variations in the data? Motivated by work in object recognition [56], recent approaches have attempted to decompose an action or event into a set of discriminative "parts" or spatio-temporal "patches" designed to capture the local spatio-temporal structure of the data [59,72]. However, these approaches still focus on the problem of classification and cannot establish strong correspondence or explain why a video is classified as a member of certain class.

Our approach is similar in spirit to work on poselets in object recognition [57]. The key idea is that instead of using semantic parts/constituents, videos are represented in terms of discriminative spatio-temporal patches that can establish correspondences across videos. However, learning poselets requires key-point annotation, which is very tedious for videos. Furthermore, for general videos it is not even clear what should actually be labeled. Recent approaches have tried to circumvent the key point annotation problem by using manually-labeled discriminative regions [73] or objectness criteria [74] to create candidate discriminative regions. This step is followed by latent models to learn the importance of candidate regions. Instead, we build upon the recent work of Singh et al. [58] and extract "video poselets" from just action labels. We do not use any priors (such as objectness) to select discriminative patches; rather we let the data select the patches of appropriate scale and translation. Also, note that our approach is different from multiple instance learning [75] since we do not assume that there exists a consistent spatio-temporal patch across positive examples (positive instance in the bag); instead we want to extract multiple discriminative patches per action class depending on the style in

which action is performed.

Outline: We first discuss how we mine the set of discriminative patches and form a vocabulary in Section 4.3. Once we have a vocabulary of discriminative patches, we use them to generate feature vectors for action classification (Section 4.4.1). Finally, we discuss how we can select correspondences based on discriminative patches using an integer programming framework (Section 4.4.2).

4.3 Mining Discriminative Patches

Given a set of training videos, we first find discriminative spatio-temporal patches which are representative of each action class. These patches satisfy two conditions: 1) they occur frequently within a class; 2) they are distinct from patches in other classes. The challenge is that the space of potential spatio-temporal patches is extremely large given that these patches can occur over a range of scales. And, the overwhelming majority of video patches are uninteresting, consisting of background clutter (track, grass, sky etc).

One approach would be to follow the bag-of-words paradigm: sample a few thousand patches, perform k-means clustering to find representative clusters and then rank these clusters based on membership in different action classes. However, this has two major drawbacks: **(a) High-Dimensional Distance Metric:** K-means uses standard distance metrics such as Euclidean or normalized cross-correlation. These standard distance metrics do not work in high-dimensional spaces [76] (In our case, we use HOG3D [77] to represent each spatio-temporal

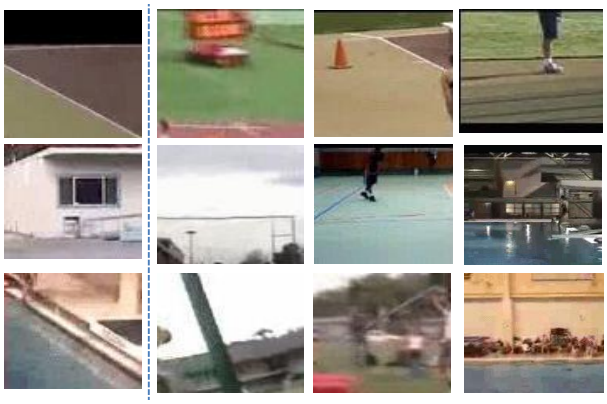


Figure 4.3: Retrieval using Euclidean Distance. (Left) Query spatio-temporal patch. (Right) Retrieval using euclidean distance metric.

patch and the dimensionality of the feature space is 1600). For example, Figure 4.3 shows a query patch (left) and similar patches retrieved using Euclidean distance (right). The Euclidean distance fails to retrieve visually similar patches. Instead, we learn a discriminative distance metric to retrieve similar patches and, hence, representative clusters. **(b) Partitioning:** Standard clustering algorithms partition the entire feature space. Every data point is assigned to one of the clusters during the clustering procedure. However, in many cases, assigning cluster memberships to rare background patches is hard. Due to the forced clustering they significantly diminish the purity of good clusters to which they are assigned.

We address these issues by using an exemplar-based clustering approach [78] which avoids partitioning the entire feature space. Every spatio-temporal patch is considered as a possible cluster center and we determine whether or not a discriminative cluster for some action class can be formed around that patch. We use the exemplar-SVM (e-SVM) approach of Malisiewicz et al. [61] to learn a dis-

criminative distance metric for each cluster. However, learning an e-SVM for *every* spatio-temporal patch in the training dataset is computationally infeasible; instead, we use motion based sampling to generate a set of initial cluster centers and then use simple nearest neighbor verification to prune candidates. The following section presents the details of this algorithm.

4.3.1 Our Approach

Available training data is partitioned into training and validation sets. The training partition is used to learn a discriminative distance metric and form clusters and the validation partition is used to rank the clusters based on representativeness. We sample a few hundred patches from each video in the training partition as candidates. We bias the sampling to avoid background patches - patches with uniform or no motion should be rejected.

However, learning an e-SVM for all the sampled patches is still computationally infeasible (Assuming 50 training videos per class and 200 sampled patches, we have approximately 10K candidate patches per class). Therefore, we perform pruning using a simple nearest-neighbor approach. For each spatio-temporal patch, we determine its k (=20, typically) nearest neighbors in the training partition. We score each patch based on how many nearest neighbors are within class as opposed to the number out of class. Based on this ranking, we select a few hundred patches per action class and use the e-SVM to learn patch-specific discriminative distance metrics. These e-SVMs are then used to form clusters by retrieving similar patches

from the training and validation partitions. Finally, we re-rank the clusters using the approach described next.

4.3.2 Ranking

Our goal is to select a smaller dictionary (set of representative patches) from the candidate patches for each class. Our criteria for ranking consists of two terms: **(a) Appearance Consistency:** We use the SVM classifier confidence as the measure of appearance consistency. The consistency score is computed by summing up the SVM detection scores of the top (10) detection scores from the validation partition. **(b) Purity:** To represent the purity/discriminativeness of each cluster we use tf-idf scores: the ratio of how many patches it retrieves from videos of the same action class to the number of patches retrieved from videos of different classes.

All patches are ranked using a linear combination of the two scores. Figure 4.4 shows a set of top-ranked discriminative spatio-temporal patches for different classes selected by this approach. As the figure shows, our spatio-temporal patches are quite representative of various actions. For example, for discus-throw, our approach extracts the patch corresponding to the turning motion before the throw (see 1st column and 2nd row) and for pull-ups it extracts the up-down motion of the body (see 1st column, 1st row). As expected, our discriminative patches are not always semantically meaningful (similar to poselets). Also, notice how our clusters exhibit good visual correspondences, which can be exploited for label transfer.

Figure 4.4: Examples of mined discriminative spatio-temporal patches that were highly ranked.

4.4 Analyzing Videos

4.4.1 Action Classification

We first evaluate our discriminative patches for action classification. We select the top n e-SVM detectors from each class and apply them in a sliding cuboid fashion to a test video. Similar to object-bank [79], we construct a feature vector based on the results of the e-SVMs. We divide each video into a hierarchical 2-level grid and spatially max-pool the SVM scores in each cell to obtain the feature vector for a video. We then learn a discriminative SVM classifier for each class using the features extracted on the training videos.

4.4.2 Beyond Classification: Explanation via Discriminative Patches

We now discuss how we can use detections of discriminative patches for establishing correspondences between training and test videos. Once a strong correspondence is established and the videos are aligned, we can perform a variety of other tasks such as object localization, finer-level action detection, etc. using simple label transfer (see Figure 4.6).

Our vocabulary consists of hundreds of discriminative patches; many of the corresponding e-SVMs fire on any given test video. This raises a question: which detections to select for establishing correspondence. One could simply use the SVM scores and select the top-scoring detections. However, individual e-SVM detections can lead to bad correspondences. Therefore, we employ a context-dependent approach to jointly select the e-SVM detections across an video. We formulate a global cost function for selection of these detections and use relaxed integer programming to optimize and select the detections.

Context-dependent Patch Selection: For simplicity, we consider the top detection of each e-SVM as a candidate detection for selection, although the approach can be extended to allow multiple (but bounded) numbers of firings of any patch. Therefore, if we have a vocabulary of size N, we have N possible candidate detections ($\{D_1, D_2, \ldots, D_N\}$) to select from. For each detection D_i, we associate a binary variable x_i which represents whether or not the detection of e-SVM i is selected. Our goal is to select the subset of detections which: (a) have high activation score (SVM score); (b) are consistent with the classified action; (c) are mutually consis-

tent. We first classify the video using the methodology described in Section 4.4.1. If our inferred action class is l, then our goal is to select the x_i such that the cost function $\mathcal{J}_l$ is minimized.

$$\mathcal{J}_l = -\sum_i A_i x_i - w_1 \sum_i C_{li} x_i + w_2 \sum_{i,j} x_i P_{ij} x_j \tag{4.1}$$

where A_i is the zero centered normalized svm score for detection i, C_{li} is the class-consistency term which selects detections consistent with action class l and P_{ij} is the penalty term which encourages selection of detections which are consistent and discourages simultaneous detections from e-SVMs which are less likely to occur together. We explain each term in detail:

- **Appearance term:** A_i is the e-SVM score for patch i. This term encourages selection of patches with high e-SVM scores.
- **Class Consistency:** C_{li} is the class consistency term. This term promotes selection of certain e-SVMs over others given the action class. For example, for the weightlifting class it prefers selection of the patches with man and bar with vertical motion. We learn C_l from the training data by counting the number of times that an e-SVM fires for each class.
- **Penalty term:** P_{ij} is the penalty term for selecting a pair of detections together. We penalize if: 1) e-SVMs i and j do not fire frequently together in the training data; 2) the e-SVMs i and j are trained from different action classes. We compute co-occurrence statistics of pairs of eSVMs on the training data to compute the penalty.

Optimization: The objective function results in an Integer Program which is an NP-hard problem. For optimizing the cost function, we use the IPFP algorithm proposed in [49]. IPFP algorithm is very efficient and the optimization converges in 5-10 iterations. IPFP solves quadratic optimization functions of the form:

$X^* = \text{argmax}(X_n^T M X_n) \quad \text{s.t. } 0 \leq X_n \leq 1$

To employ IPFP, we transform the cost function to the above form through the following substitution: $X_n = \begin{pmatrix} 1 \\ X \end{pmatrix}$ and $M = \begin{pmatrix} 1 & \frac{(A+C)^T}{2} \\ \frac{(A+C)}{2} & -P \end{pmatrix}$.

The solution obtained by the IPFP algorithm is generally binary, but if the output is not binary then we threshold at 0.5 to binarize it. The set of patches which maximizes this cost function is then used for label transfer and to infer finer details of the underlying action.

4.5 Experimental Evaluation

We demonstrate the effectiveness of our representation for the task of action classification and establishing correspondence. We will also show how correspondence between training and test videos can be used for label transfer and to construct detailed descriptions of videos.

Datasets: We use two benchmark action recognition datasets for experimental evaluation: UCF-50 and Olympics Sports Dataset [65]. We use UCF-50 to qualitatively evaluate how discriminative patches can be used to establish correspondences and transfer labels from training to test videos. We manually annotated the videos in 13 of these classes with annotations including the bounding boxes of

objects and humans (manually annotating the whole dataset would have required too much human effort). We also performed a sensitivity analysis experiment (w.r.t. to vocabulary size) on this subset of UCF-50 dataset. Quantitatively, we evaluate the performance of our approach on action classification on the UCF-50 and the complete Olympics dataset.

Implementation Details: Our current implementation considers only cuboid patches, and takes patches at scales ranging from 120x120x50 to the entire video. Patches are represented with HOG3D features (4x4x5 cells with 20 discrete orientations). Thus, the resulting feature has 4x4x5x20 = 1600 dimensions. At the initial step, we sample 200 spatio-temporal patches per video. The nearest neighbor step selects 500 patches per class for which e-SVMs are learned. We finally select a vocabulary of 80 e-SVMs per class. During exemplar learning, we use a soft-margin SVM with C fixed to 0.1. The SVM parameters for classification are selected through cross validation.

4.5.1 Classification Results

UCF Dataset: The UCF50 dataset can be evaluated in two ways: videowise and groupwise. We tested on the more difficult task of groupwise classification guaranteeing that the backgrounds and actors between the training and test sets are disjoint. We train on 20 groups and test on 5 groups. We evaluate performance by counting the number of videos correctly classified out of the total number of videos in each class. Table 4.1 shows performance of our algorithm compared to the action bank approach [1] on 13 class subset (run with same test-train set as our approach)

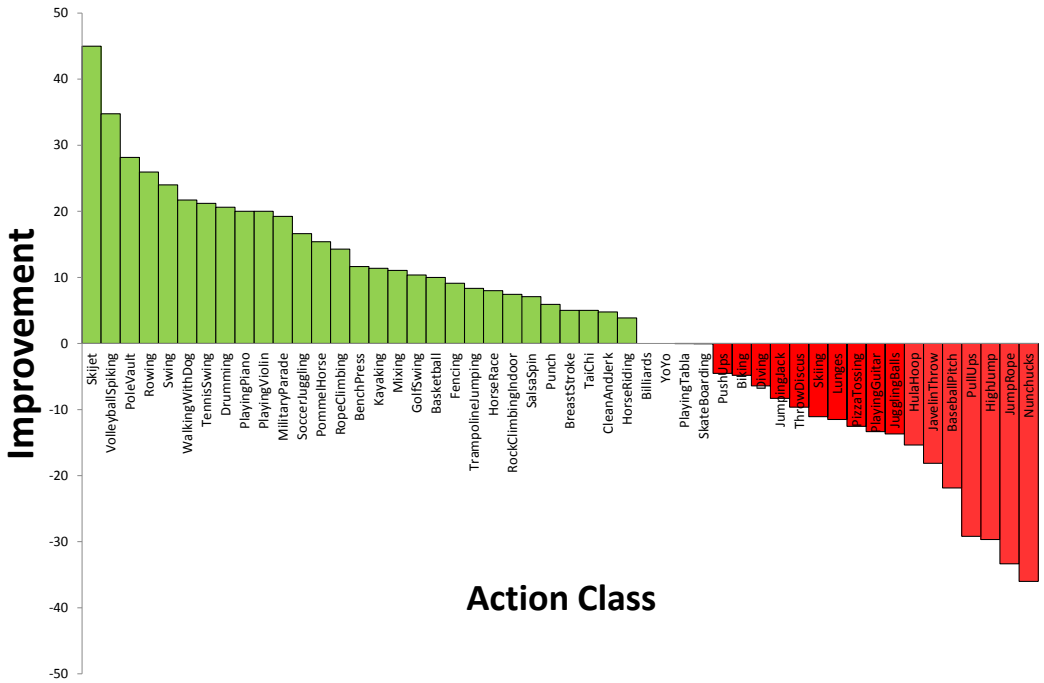


Figure 4.5: Improvement in performance per action class comapared to [1]

and a bag-of-words approach as a baseline. We also evaluated performance with respect to vocabulary size. Table 4.4 shows the performance variation with the number of e-SVM patches trained per class. Finally, we evaluate action classification for all 50 classes in UCF and we get an improvement of 3.3% over [1]. Table 4.5 shows quantitative performance and Figure 4.5 compares performance with [1] for each of the 50 classes.

Olympics Dataset: We follow the same experimental setting for splitting the data into test-train and employ the same evaluation scheme (mAP) as used in [59]. Tables 4.2 and 4.3 show the performance of our approach versus previous approaches.

Action Class	BoW(baseline)	[1]	Ours
Basketball	20.00	**53.84**	50.00
Clean and Jerk	40.00	85.00	**95.65**
Diving	58.06	**78.79**	61.29
Golf Swing	54.84	**90.32**	75.86
High Jump	12.90	38.46	**55.56**
Javeline Throw	29.03	45.83	**50.00**
Mixing	12.90	42.85	**55.56**
PoleVault	65.62	60.60	**84.37**
Pull Up	48.88	**91.67**	75.00
Push Ups	40.63	85.00	**86.36**
Tennis Swing	**51.51**	44.12	48.48
Throw Discus	63.64	75.00	**87.10**
Volleyball Spiking	24.24	43.48	**90.90**
Mean Classification	40.17	64.23	**70.47**

Table 4.1: Classification performance of our algorithm compared to Action Bank [1] in groupwise division of dataset

Sport Class	[59]	[80]	[81]	Ours
High-jump	68.9	52.4	75.8	**84.94**
Long-jump	74.8	66.8	78.6	**84.64**
Triple-jump	52.3	36.1	69.7	**83.29**
Pole-vault	82.0	47.8	85.5	**84.67**
Gymnastics-Vault	86.1	88.6	**89.4**	82.58
Shot-put	62.1	56.2	65.9	**83.55**
Snatch	69.2	41.8	72.1	**83.47**
Clean-jerk	84.1	83.2	86.2	**86.64**
Javelin-throw	74.6	61.1	77.8	**84.75**
Hammer-throw	77.5	65.1	79.4	**86.40**
Discus-Throw	58.5	37.4	62.2	**86.66**
Diving-platform-10m	87.2	**91.5**	89.9	86.51
Diving-springboard-3m	77.2	80.7	82.2	**86.44**
Basketball-layup	77.9	75.8	79.7	**88.60**
Bowling	72.7	66.7	78.7	**88.27**
Tennis-serve	49.1	39.6	63.8	**83.37**

Table 4.2: Quantitative Evaluation on Olympics Sports Dataset. Mean results are shown in the next table.

Approach	mAP
Niebles et. al. [59]	71.1
Laptev et. al. [80]	62.0
William et. al. [81]	77.3
Adrien et. al. [82]	82.7
Ours	85.3

Table 4.3: Comparison on Olympics Dataset

Patches per class	mAP
20	62.52
30	65.25
50	67.57
80	70.47
100	70.17

Table 4.4: Effect of Vocabulary Size on UCF13

4.5.2 Correspondence and Label Transfer

We now demonstrate how our discriminative patches can be used to establish correspondence and align the videos. Figure 4.6 shows a few examples of alignment using the detections selected by our framework (additional examples are shown in the supplementary material). It can be seen that our spatio-temporal patches are insensitive to background changes and establish strong alignment. We also use the aligned videos to generate annotations of test videos by simple label-transfer technique. We manually labeled 50 discriminative patches per class with extra annotations such as objects of interaction (e.g, weights in clean-and-jerk), person bounding boxes and human poses. After aligning the videos we transfer these annotations to the new test videos.

Figure 4.6 shows the transfer of annotations. These examples show how strong correspondence can allow us to perform tasks such as object detection, pose esti-

BaseballPitch	37.5	BasketBall	60.0	BenchPress	94.0
Biking	40.0	Billards	100	BreastStroke	100
CleanAndJerk	70.0	Diving	71.0	Drumming	47.1
Fencing	77.3	GolfSwing	69.0	HighJump	44.4
HorseRace	88.0	HorseRiding	90.4	HulaHoop	30.8
JavelinThrow	36.4	JugglingBalls	13.6	JumpRope	25.0
JumpingJack	88.0	Kayaking	57.1	Lunges	30.8
MilitaryParade	57.7	Mixing	40.7	Nunchucks	0
PizzaTossing	20.8	PlayingGuitar	60.0	PlayingPiano	95.0
PlayingTabla	65.2	PlayingViolin	50.0	PoleVault	84.4
PommelHorse	73.1	Pullup	45.8	Punch	90.3
PushUps	59.1	RockClimbingIndoor	77.8	RopeClimbing	60.0
Rowing	70.4	SalsaSpin	57.1	SkateBoarding	61.5
Skiing	59.3	Skijet	85.0	SoccerJuggling	53.3
Swing	64.0	TaiChi	55.0	TennisSwing	51.5
ThrowDiscus	71.0	TrampolineJumping	75.0	VolleyballSpiking	91.3
WalkingWithDog	47.8	YoYo	62.5		

Table 4.5: Quantitative Evaluation on UCF50 dataset

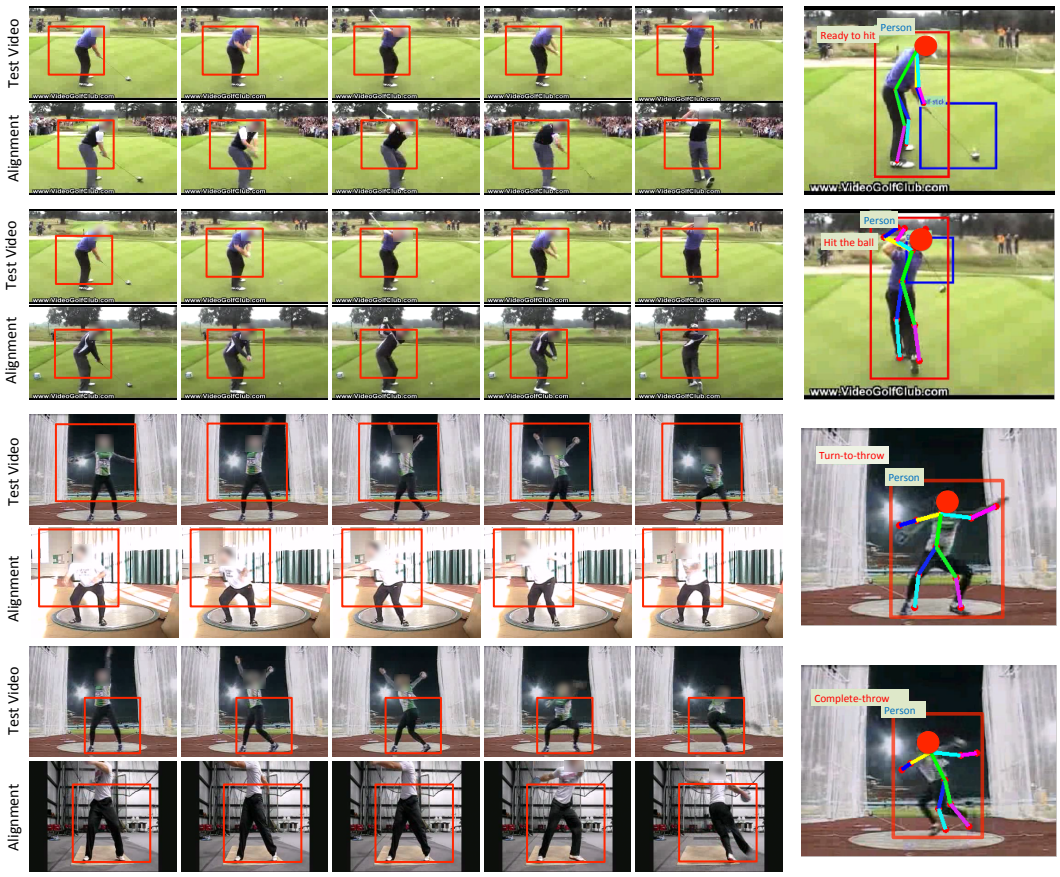


Figure 4.6: Rich Annotations using Label Transfer: We show how discriminative patches help us to align test video with training videos. After the videos are aligned we use them to obtain rich annotations such as object bounding boxes and human poses by simple label transfer.

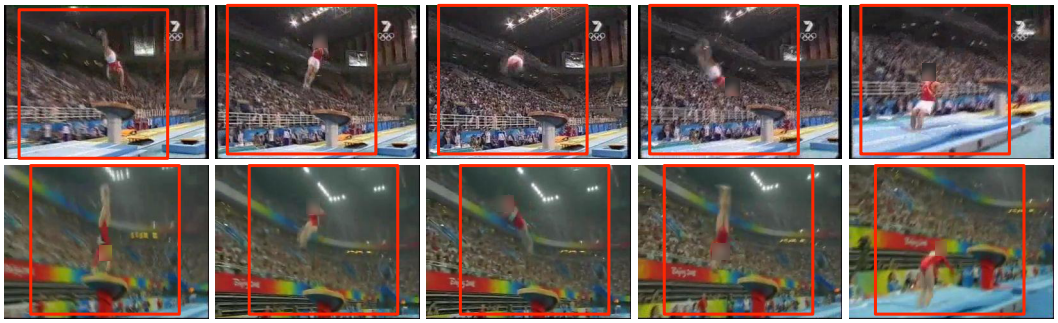


Figure 4.7: Example alignment in case of Olympics Dataset. Additional examples are shown in the supplementary material.

mation and predicting temporal extent. For example, detecting the golf club in the golf-swing case is extremely difficult because the golf club occupies very few pixels in the video. But our strong alignment via motion allows us to transfer the bounding box of the golf-club to the test video. Similarly, estimating human poses for golf-swing and discus throw would be extremely difficult. But again, using our discriminative spatio-temporal patches we just align the videos using motion and appearance and then transfer the poses from the training videos to test videos. We also did an informal evaluation of our pose transfer. For 50 randomly sampled transfers, more than 50% of the transferred joints are within 15 pixels of the ground-truth joint locations.

Chapter 5: Text Detection and Recognition in Natural Scenes and Consumer Videos

5.1 Introduction

An end-to-end system for text detection and recognition is important in multiple domains such as content based retrieval systems, video event detection, human computer interaction, autonomous robot or vehicle navigation and vehicle license plate recognition. Further there are several commercial systems for scanned document text [83] [84]. However, these systems typically need cropped and binarized text regions to perform well [85]. Text detection in natural scenes is a challenging problem and has gained a lot of attention recently [86]. Such text presents challenges because of low contrast with background, large variation in font, color, scale and orientation combined with background clutter. Therefore a robust and fast recognition system is desirable.

Text detection approaches can be divided into two main categories (a) sliding window based approaches (b) connected components based approaches. In sliding window based approaches, low-level features are extracted for each scanning window and each candidate is evaluated for presence of text using machine learning

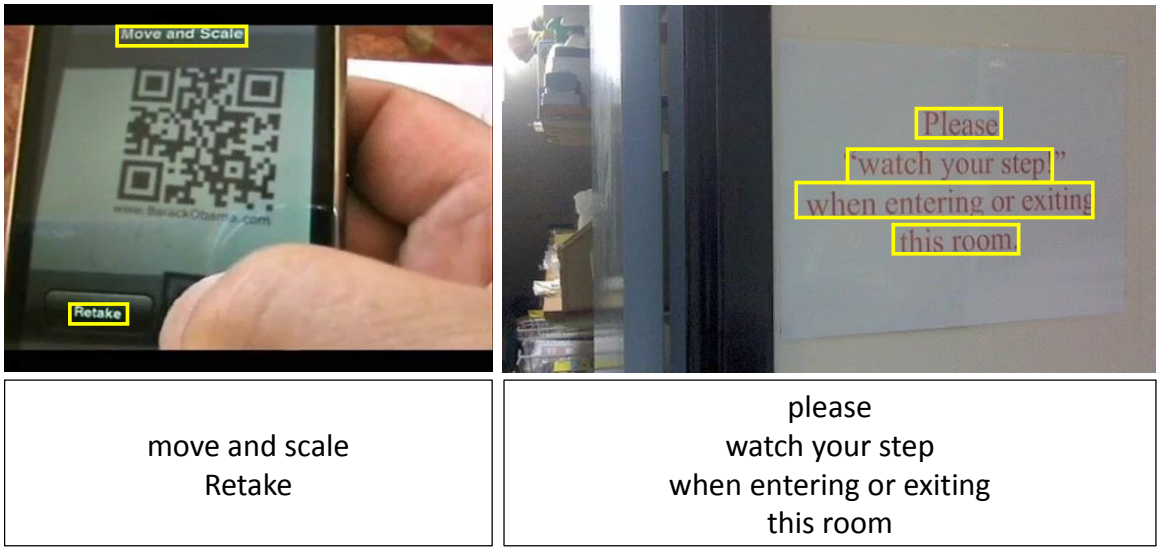


Figure 5.1: Proposed end-to-end system for text detection and recognition

techniques. [87] used Histogram of Oriented Gradient (HOG) features together with Random fern technique for text recognition. The ambiguities in recognition were fixed using a pictorial structure with lexicon to create an end-to-end system for text detection and recognition. [88] used gradient, edge, texture and Gabor features together with adaboost learning technique for classification. [89] classified text windows from non-text using principle stroke Gabor words and showed improvement over previous approaches. However, these approaches do not explicitly account for scale variations and therefore most of these approaches are applied over multiple scales and results are aggregated into single detection result.

On the other hand, Connected Component (CC) approaches first extract pixel regions which have similar edge strength, color, texture or stroke width and evaluate each one of them for being text or not text using rule-based or machine learning techniques. [90] used low-variance in text stroke width as a measure to select

text candidate regions from non-text regions. [91] extracted candidate regions using edgelinks (continuous edge chains) and evaluated each candidate using a Support Vector Machine (SVM) classifier. The output of the SVM is integrated using a Conditional Random Field (CRF). Recently, Stable Extremal Regions has become popular approach to extract connected component candidates [92] as they are robust to illumination and scale changes. [93] showed a real-time system for text detection in videos using extremal regions. [94] proposed geometric grouping over MSER regions and classified the regions using adaboost. [95] built a graph network over MSER candidates and determined text from non-text region using graph cut.

We propose an end-to-end system for text detection and recognition in video frames. The proposed system comprises of three steps (a) text localization (b) text line aggregation (c) text line recognition. We use MSER regions as candidates and instead of using rule or geometric based grouping, we apply a text/non-text SVM classifier over each candidate. We compute rich shape descriptors and compresses them to very few dimensions while preserving discriminability using Partial Least Squares (PLS) technique. PLS technique has two advantages: (a) It allows us to use a large set of discriminative features for classification. (b) It speeds up the classification step. Each positively labeled candidate serves as an anchor region around which we group candidate regions based on geometric and color properties. At this step, we allow negatively labeled candidates as well to take part in text line aggregation, to overcome mistakes of the classification step. We binarize the detected text regions and pass it to an OCR system for word recognition.

5.2 Text Localization

5.2.1 Text Candidates using MSER

MSER technique, proposed by Matas et. al. [92], finds stable connected regions over a range of thresholds. This technique was originally used for correspondence between two images with different viewpoints. Low-level image segmentation as a prerequisite step of text detection can also benefit from MSER. MSER is able to detect most of the characters even in low resolution video frames. We prune candidates of sizes smaller than a predefined threshold t_l or larger than t_h. We also prune candidates of aspect ratios outside the range $[r_l,r_h]$, and with numbers of holes beyond a threshold h_{th}. After MSER candidates are extracted, we compute features for training a text/non-text SVM classifier.

5.2.2 Feature Extraction

We extract a rich set of features for classifying candidate regions into text or non-text (background). Histogram of Oriented Gradients (HOG), proposed by [96], showed impressive result for object and human detection. With an image divided into cells, HOG features are rich shape descriptors which captures the shape of the object by quantizing the gradient information in each cell. These cells are grouped into equal or larger sized overlapping blocks which are then normalized and concatenated together to form the feature vector. Figure 2 shows visualization of HOG feature for letter A and X. Each cell is represented by an oriented "star"

showing the strength of corresponding gradient direction. In order to keep the feature dimensions consistent, all the text candidate regions are resized to fixed size before computing HOG features.

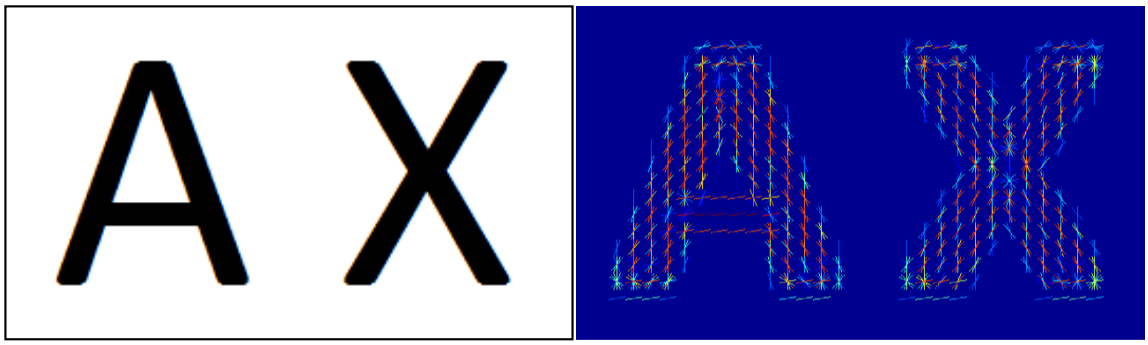


Figure 5.2: Visualization of HOG features

Gabor filter is a band-pass filter which can be viewed as a sinusoidal plane of particular frequency and orientation modulated by a Gaussian function. It extracts orientation-dependent frequency information such as direction of strokes which can be used to discriminate text from non-text. We use the standard deviation on output of Gabor filters on candidate regions as feature. The 2-D Gabor filter can be written in the following form:

$$G(x, y, \lambda, \phi, \sigma_x, \sigma_y) = \frac{1}{2\pi\sigma_x\sigma_y} \exp\left\{\frac{1}{2}\left(\frac{R_1^2}{\sigma_x^2} + \frac{R_2^2}{\sigma_y^2}\right)\right\} \exp\left\{i\frac{2\pi R_1}{\lambda}\right\}$$

where $R_1 = xcos\phi + ysin\phi$ and $R_2 = ycos\phi - xsin\phi$, λ is the wavelength of Gabor filter, ϕ is the orientation of Gabor filter and σ_x and σ_y denotes the standard deviation of Gabor filter. For simplicity, $\sigma_x = \sigma_y = \sigma$.

We compute the ratio of corners to edges [91] for each text region according

to the following equation,

$$\xi = \sum_{x=1}^{w}\sum_{y=1}^{h} C(x,y) / \sum_{x=1}^{w}\sum_{y=1}^{h} E(x,y), \tag{5.1}$$

where $w \times h$ is the size of bounding box for the text region, $C(x,y)$ denotes the intensity of corner obtained after binarization with a fixed threshold of Harris corner detection [97] result over the frame, $E(x,y)$ denotes the intensity of edge map of the input image obtained using Canny Edge detection algorithm [98].

5.2.3 Dimensionality reduction using PLS

Speed is an important factor when we are building a practical system for text detection and recognition in videos. We found that SVM classification on original features is the bottleneck in computational efficiency. Hence, we apply PLS technique for dimensionality reduction, compressing the original feature space ($\sim$2300 dimensions) to just few dimensions (9 dimensions) while preserving the discriminability. This gives us 5x speed up, which is significant given the size of our dataset. We briefly describe mathematical formulation of PLS technique below. More detailed discussion can be found in [99].

Let $X_{n\times m} \subset \mathbb{R}^m$ denote an m dimensional feature vectors of sample size n and let $Y_{n\times 1} \subset \mathbb{R}$ be their corresponding 1-dimensional class labels. PLS decomposes the zero-mean matrix $X_{n\times m}$ and zero-mean $Y_{n\times 1}$ into

$$X = TP^T + E$$

$$Y = Uq^T + f \tag{5.2}$$

where T and U are $n \times p$ matrices containing p extracted latent vectors. The matrix $P_{m \times p}$ and $q_{1 \times p}$ represents the loading, similar to Principal Component Analysis (PCA). E and f represents residual error while projecting data onto lower subspace for X and Y respectively. The Nonlinear iterative partial least squares algorithm (NIPALS) [99] constructs a set of weight vectors $W = w_1, ..., w_p$ such that,

$$[cov(t_i, u_i)]^2 = \max_{|w_i=1|} [cov(Xw_i, y)]^2 \tag{5.3}$$

where t_i is the i-th column of T matrix, u_i the i-th column of matrix U, and $cov(t_i, u_i)$ is the covariance between latent vector t_i and u_i.

PLS find subspaces where the covariance between projected feature X and label Y is maximized. Therefore, a key difference between PCA and PLS is that PLS exploits label information while finding latent subspace while PCA does not. The resultant W matrix is used to project data into the low dimensional subspace which is used to learn SVM classifier on training data and classify text from non-text regions during testing.

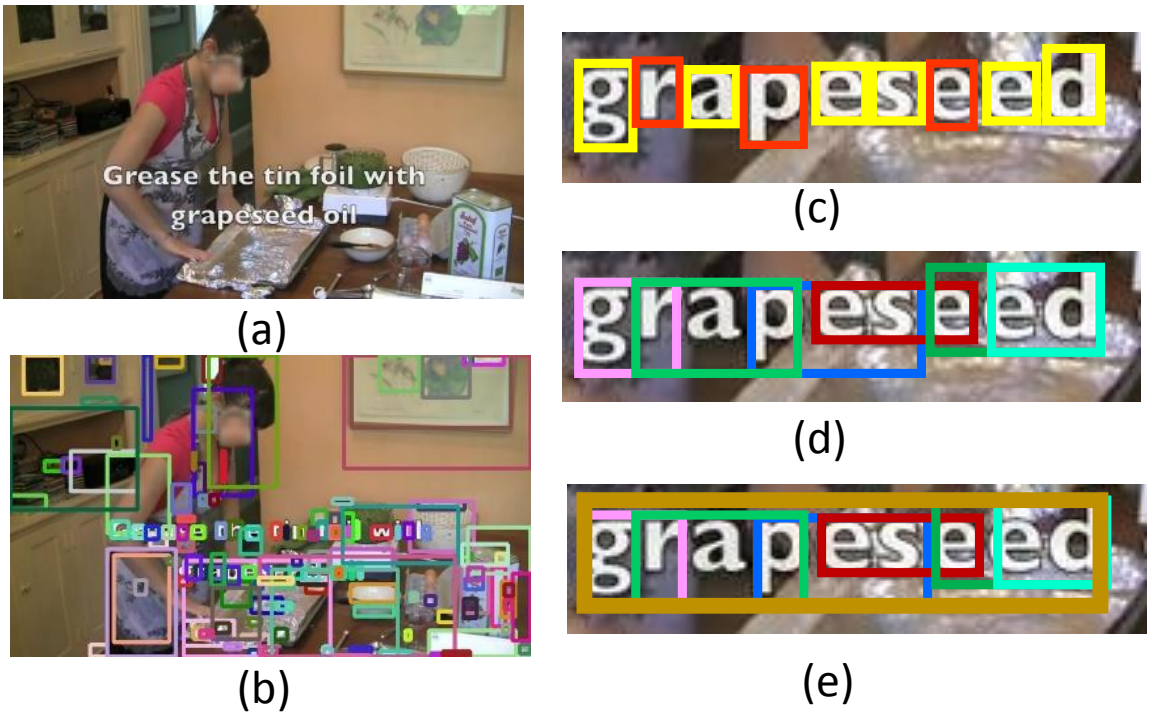


Figure 5.3: (a) original image, (b) MSER candidates, (c) SVM classifier result (positive in yellow and negative in red), (d) *grouplets* after merging (each *grouplet* showed in different color), (e) detected text bounding box

5.2.4 SVM classifier

We extract MSER regions from training data, which are separated into positive and negative data according to manual annotation of text bounding boxes in the data. A region is considered positive only if it overlaps more than 90% with a ground truth bounding box. We extract features from each region and project them onto a lower PLS subspace. We then learn a SVM classifier in the projected subspace [46].

5.3 grouping of localized text regions

Each positively classified MSER region serves as an anchor for grouping the text into words during testing. We want to emphasize at this point that the previous classification step is only used to localize the regions with high text probabilities. MSER regions misclassified by the SVM will still be considered for potential merging with these anchors if they satisfy certain criteria. This procedure will allow us to overcome the mistakes of the classification step when hypothesizing grouped text regions.

For each positively classified MSER region, we search its neighborhood for MSER regions which have similar color, size, aspect ratio and proximate enough to form a word. At this step, we consider all the initial candidates irrespective of their classification label. If a MSER region satisfies the criterion for merging, then the anchor and the searched regions are merged into a '*grouplet*'. Each positive anchor can at most connect to two adjacent regions and a single region can be part of multiple *grouplets* (Figure5.3(d)). If an anchor does not connect to any neighboring region, then it is discarded. All the regions which do not merge are also discarded from further analysis.

We then follow a simple heuristic scheme to merge these *grouplets* into words. Two *grouplets* will be merged if they are spatially close and if they have similar color, height and aspect ratio. This step is continued until no other *grouplet* can be merged with one another. The bounding box obtained after all overlap and merging is considered the final result.

Figure5.3 shows the steps of text grouping. Even though the letters "r", "p" and "e" are misclassified as non-text by SVM, they get merged into different *grouplets* and become part of final detection result.

5.4 OCR decoding

We pre-process each text line before OCR decoding. Each cropped textline image is first binarized using Otsu method [100]. When the text has a dark background and a bright foreground, the image is inverted before the thresholding is applied. A median filter is applied to remove salt and pepper noise. Finally, the image is resized to fixed height of 110 with its aspect ratio unchanged and passed for OCR decoding.

5.4.1 OCR system

We use the BBN HMM Byblos OCR system for decoding [101]. We train the OCR system using text from printed English documents. We briefly describe the mathematical model used in OCR system as follows. Lets assume that text line is represented by a sequence of feature vectors X. The goal is to find sequence of characters (C) that best explain the features X. Mathematically this can be written as $P(C|X)$ which when expanded using Bayes' rule,

$$P(C|X) = \frac{P(X|C)P(C)}{P(X)}, \tag{5.4}$$

where $P(X|C)$ is the model learned from training data. $P(C)$, the language model, is the prior probability for allowed sequence of characters. The language model

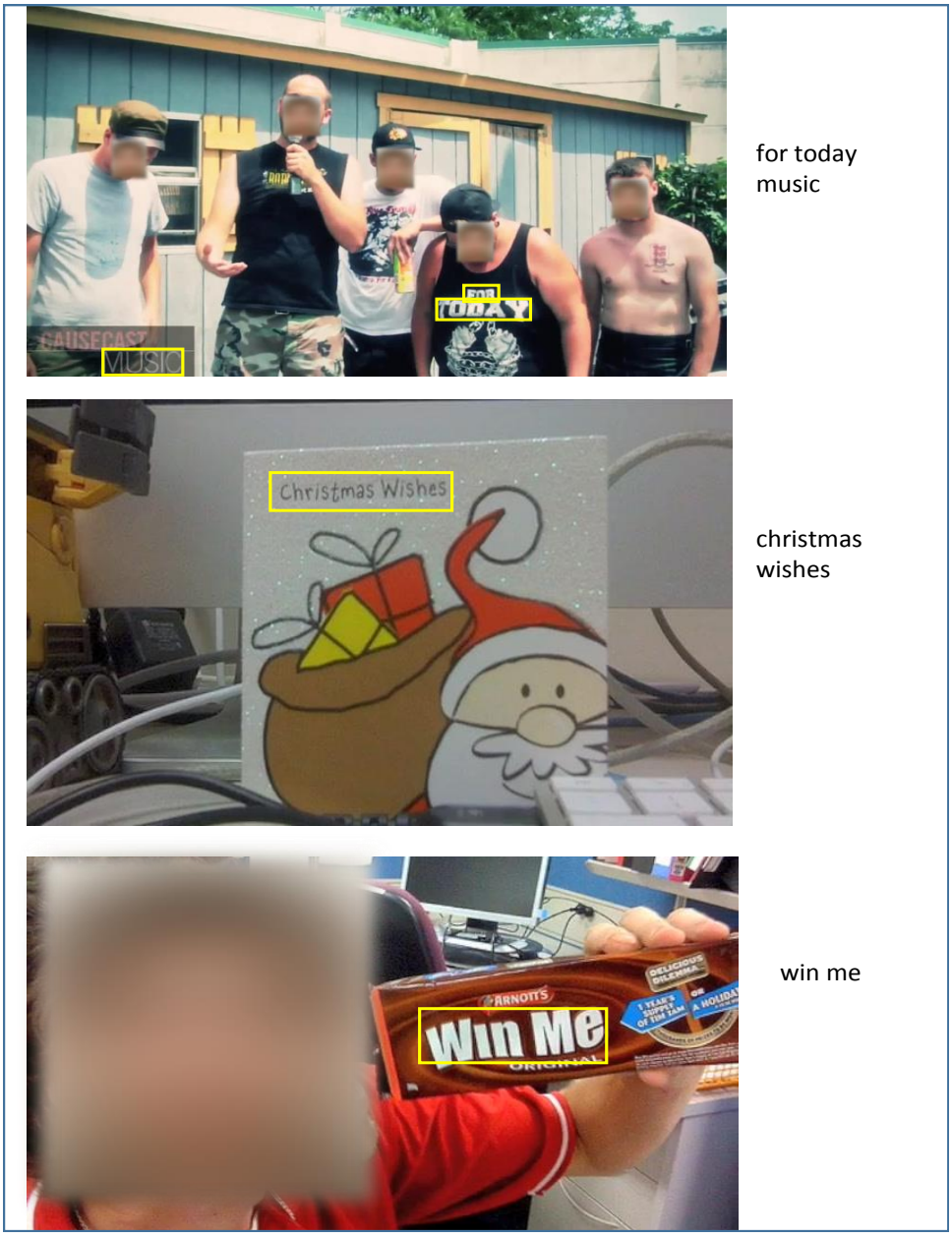


Figure 5.4: Qualitative results of our algorithm

used in the OCR system is a finite-lexicon word n-gram Markov model. The goal is to maximize likelihood term, $P(X|C)P(C)$ since $P(X)$ is independent of C. More details about the OCR engine can be found in [101].

5.5 Experiments

We evaluated the performance of our system on an extremely large consumer video dataset. We divided the evaluations into two tasks.

Task 1) Text Detection and Recognition: We selected a subset of 1750 videos from the TRECVID MED dataset [102], which is created from consumer videos on web. Each video frame is annotated for text region bounding boxes and underlying words. These are unconstrained videos with varying backgrounds, text fonts and stroke widths, which make them extremely challenging for text detection and recognition.

Implementation details: For our experiments, cell size and block size are set to 4 pixels and each candidate is resized to 32×32 before computing HOG features, resulting in 2255 feature dimensions. Gabor filters are computed for $\phi = \left\{0, \frac{\pi}{4}, \frac{\pi}{2}, \frac{3\pi}{4}\right\}$, $\lambda = \{1, 2, 4, 10\}$ and $\sigma = \{2, 4\}$ values, resulting in 32 dimensional Gabor features.

We sampled video frames uniformly at the rate of 2 fps and ran our text detection and recognition system. We compare the OCR output performance based on the proposed text detector with that based on the CRF based detector [91] in Table 5.1. We significantly improve both frame-level word precision and recall scores for OCR output compared to [91].

In order to evaluate pixel level precision-recall for our text detection algorithm, we collected 596 images from these videos and divided them into 388 train and 208 test images. We followed the same evaluation scheme described in [91] to

Table 5.1: OCR Word Recognition Performance

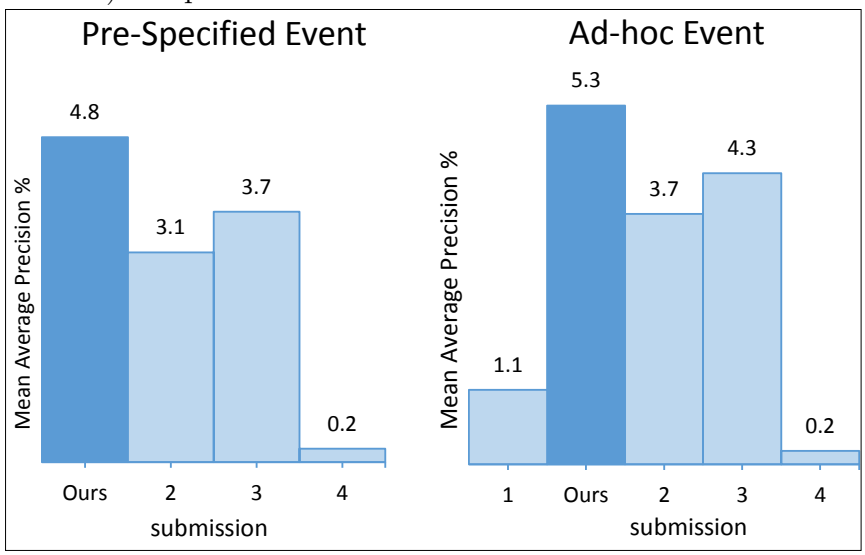
	Precision	Recall	F-score
[91]	0.045	0.234	0.076
Ours	0.147	0.370	**0.210**

Table 5.2: Text Detection Performance

	Precision	Recall	F-score
[91]	0.7066	0.1444	0.2392
Ours	0.5209	0.3024	**0.3823**

compute precision-recall scores. Table 5.2 shows the performance comparison for text detection.

Figure 5.5: Performance of BBN's OCR-only system on TRECVID MED task (Ek100 condition) compared with other submissions.



Task 2) Event Detection: We also include TRECVID Multimedia Event

Detection (EK100 condition) [102] performance, based on only the video text content. The dataset consists of about 100,000 consumer-generated videos, including occurrences of 20 pre-specified events and 10 additional ad-hoc events. The SVM classifier for each event is trained with 100 positive videos and a set of background videos. The BBN OCR-only system uses the video text detection and recognition output from the components described in this paper. The OCR decoding output word lattice for each video is converted into a vector where each dimension corresponds to one different word weighted by its expected count in the lattice and inverse document frequency. Table 5 illustrates the OCR-only Multimedia Event Detection (MED) performance measures in the TRECVID 2013 MED evaluation, highlighting the competitiveness of our system.

Chapter 6: Conclusion and Future Research Direction

6.1 Scene dependent contextual modeling

We show that contextual information is scene dependent and not all contextual information is equally important. We present a data-driven approach to learn "what" contextual information is useful and "how" they should be incorporated in scene understanding. Our iterative approach jointly learns importance of edges in the Markov Network and contextual feature weights associated with each edge based on statistical models of global and local image features. Experimental results show that this scene dependent Markov Network eliminates spurious edges and improves performance over fully-connected and neighborhood connected Markov networks.

In future, we would like to investigate schemes to speed up the training process. Currently, training process is time consuming as we need to learn edge weights and contextual importance for each image in the training dataset.

6.2 Incorporating context in a multiple segmentation and recognition framework

We describe an approach for a multiple segmentation framework and labeling of images using appearance and context. In order to overcome the fragmentation of object by generic segmentation algorithms, we propose a merging function which merges adjacent segments to increase the spatial support of segments. These segments are then arranged in a hierarchical fashion and path constrained are added for segment selection. The optimization function, comprising of appearance and context term, was solved by relaxing the discrete constraints and employing a Quadratic Programming method. The relaxed solution is then discretized using a greedy algorithm. Experiments on three well studied datasets demonstrate the advantages of the method.

With the recent surge in photos and videos taken from hand-held devices and those shared online, many of which are taken in the same scenes, the need to automatically label objects in such images has emerged. We are currently investigating extending this work for labeling multiple images of same scenes simultaneously. Preliminary results show that our approach, when extended to label multiple images simultaneously, improves labeling performance compared to labeling each image individually.

6.3 Discriminative patch based representation of videos

We propose a new representation for videos based on discriminative spatio-temporal patches. Unlike previous works, we don't enforce semanticity on these patches. These spatio-temporal patches might correspond to a human action, a semantic object, human-object pair or perhaps a random but informative spatio-temporal patch in the video. We also showed how exemplar based clustering preserves the purity of clustering opposed to other clustering approaches such as k-means which partitions the entire feature space. We automatically mine these patches from hundreds of training videos using exemplar-based clustering approach. In order to speed up the process, we use nearest neighbor scheme to select candidates before learning exemplar-SVM on them. We have also shown how these patches can be used to obtain strong correspondence and align the videos for transferring annotations. Furthermore, these patches can be used as a vocabulary to achieve state of the art results for action classification. Our framework is generic and can be applied to similar problems in other fields where the goal is to find discriminative elements in the data.

In future, we want to investigate methods to speed up the exemplar-SVM training step. A recent work [103] used mode-seeking algorithm to mine these discriminative patches which completely by-passes expensive exemplar-SVM step. We also want to explore graph matching algorithms for better alignment of patches.

6.4 Text Detection and Recognition for event detection in consumer videos

We propose an end-to-end text detection and recognition system. Maximally stable extremal regions and contours based on connected edges are considered as candidates for text detection. The text detection component uses SVM classifier based on rich shape descriptors such as HOG, Gabor and edge features to classify text from non-text candidates. We use Partial Least Squares technique for dimensionality reduction which leads to 5x speed improvement. Our proposed merging scheme overcomes the mistakes of SVM classification step and preserves word boundaries. Extensive evaluation on a large dataset confirms that our approach significantly outperforms other approaches in both pixel-level text detection and word recognition tasks. Furthermore, the event detection system built upon the OCR output of this approach outperformed multiple other OCR-only based submissions in the recently concluded NIST TRECVID 2013 multimedia Ek100 event detection evaluations.

Chapter A: List of Published and Submitted Publications

Arpit Jain, Xujun Peng, Xiaodan Zhuang, Pradeep Natarajan and Huaigu Cao, Text Detection and Recognition in Natural Scenes and Consumer Videos, In 39th International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2014.

Arpit Jain, Abhinav Gupta, Mikel Rodriguez, Larry S Davis, Representing Videos using Mid-level Discriminative Patches, IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2013

Arpit Jain*, Stephen Xi Chen*, Abhinav Gupta, Larry S Davis, Piecing Together the Segmentation Jigsaw using Context, IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2011 (* equal contribution)

Arpit Jain, Abhinav Gupta, Larry S Davis, Learning What and How of Contextual Models for Scene Labeling, European Conference on Computer Vision (ECCV) 2010.

Bibliography

[1] Sreemanananth Sadanand and Jason J. Corso. Action bank: A high-level representation of activity in video. In *IEEE Conference on Computer Vision Pattern Recognition*, 2012.

[2] P. Carbonetto, N. Freitas, and K. Barnard. A statistical model for general contextual object recognition. In *European Conference on Computer Vision*. 2004.

[3] Abhinav Gupta and Larry Davis. Beyond nouns: Exploiting prepositions and comparative adjectives for learning visual classifiers. In *European Conference on Computer Vision*. 2008.

[4] C. Galleguillos, A. Rabinovich, and S. Belongie. Object categorization using co-occurrence, location and appearance. In *IEEE Conference on Computer Vision Pattern Recognition*. 2008.

[5] Geremy Heitz and Daphne Koller. Learning spatial context: Using stuff to find things. In *Computer Vision–In European Conference on Computer Vision 2008*, pages 30–43. Springer, 2008.

[6] Kevin Murphy, Antonio Torralba, and William Freeman. Using the forest to see the trees: a graphical model relating features, objects and scenes. *Advances in neural information processing systems*, 16, 2003.

[7] J. Shotton, M. Johnson, and R. Cipolla. Semantic texton forests for image categorization and segmentation. In *Proc. IEEE Conference on Computer Vision Pattern Recognition*. 2008.

[8] Lior Wolf and Stanley Bileschi. A critical view of context. *International Journal of Computer Vision*, 69(2):251–261, 2006.

[9] S. Divvala, D. Hoiem, J. Hays, A. A. Efros, and M. Hebert. An empirical study of context in object detection. In *IEEE Conference on Computer Vision Pattern Recognition*. 2009.

[10] B. C. Russell, A. A. Efros, J. Sivic, W. T. Freeman, and A. Zisserman. Using multiple segmentations to discover objects and their extent in image collections. In *European Conference on Computer Vision*. 2006.

[11] Tomasz Malisiewicz and Alexei A. Efros. Recognition by association via learning per-exemplar distances. In *IEEE Conference on Computer Vision Pattern Recognition*. 2008.

[12] J. Li and L. Fei-Fei. What, where and who? classifying event by scene and object recognition. In *International Conference on Computer Vision*. 2007.

[13] X. He and R. Zemel. Latent topic random fields: Learning using a taxonomy of labels. In *IEEE Conference on Computer Vision Pattern Recognition*. 2008.

[14] K. Murphy, A. Torralba, and W. Freeman. Using the forest to see the trees:a graphical model relating features, objects and scenes. In *Conference on Neural Information Processing Systems*. 2003.

[15] A. Gupta and L. S. Davis. Objects in action:an approach for combining action understanding and object perception. In *IEEE Conference on Computer Vision Pattern Recognition*. 2007.

[16] A. Rabinovich, A. Vedaldi, C. Galleguillos, E. Wiewiora, and S. Belongie. Objects in context. In *International Conference on Computer Vision*. 2007.

[17] P. Kohli M. Szummer and D. Hoiem. Learning crfs using graph cuts. In *European Conference on Computer Vision*. 2008.

[18] Ioannis Tsochantaridis, Thorsten Joachims, Thomas Hofmann, and Yasemin Altun. Large margin methods for structured and interdependent output variables. In *Journal of Machine Learning Research*, pages 1453–1484, 2005.

[19] A. Torralba, K.P. Murphy, and W.T. Freeman. Contextual models for object detection using boosted random fields. In *Conference on Neural Information Processing Systems*. 2005.

[20] N. Friedman. The bayesian structural em algorithm. *UAI*, 1998.

[21] L.K McDowell, K. Gupta, and David. Aha. Cautious inference in collective classification. In *AAAI*. 2007.

[22] J. Neville and D. Jensen. Iterative classification in relational data. In *AAAI Workshop on Learning Statistical Models from Relational Data*. 2000.

[23] A. Rakotomamonjy, F. Bach, S. Canu, and Y. Grandvalet. More efficiency in multiple kernel learning. In *International Conference on Machine Learning*. 2007.

[24] M. Galun, E. Sharon, R. Basri, and A. Brandt. Texture segmentation by multiscale aggregation of filter responses and shape elements. In *International Conference on Computer Vision*. 2003.

[25] Bryan C. Russell, Antonio Torralba, Kevin P. Murphy, and William T. Freeman. Labelme: a database and web-based tool for image annotation. In *International Conference on Computer Vision*. 2008.

[26] A. Oliva and A. Torralba. Building the gist of a scene: The role of global image features in recognition. *Visual Perception*, 2006.

[27] D. Hoiem, A.A. Efros, and M. Hebert. Geometric context from a single image. In *International Conference on Computer Vision*. 2005.

[28] Prateek Jain and Ashish Kapoor. Probabilistic nearest neighbor classifier with active learning. *Microsoft Research, Redmond*.

[29] T. Malisiewicz and A. Efros. Beyond categories: The visual memex model for reasoning about object relationships. In *Conference on Neural Information Processing Systems*. 2009.

[30] Stephen Gould, Tianshi Gao, and Daphne Koller. Region-based segmentation and object detection. In *Conference on Neural Information Processing Systems*. 2009.

[31] M. Pawan Kumar and Daphne Koller. Efficiently selecting regions for scene understanding. In *Conference on Neural Information Processing Systems*. 2010.

[32] Abhinav Gupta, Alexei Efros, and Martial Hebert. Block world revisited: Image understanding using qualitative geometry and mechanics. In *European Conference on Computer Vision*. 2010.

[33] J. M. Tenenbaum and H. G Barrow. Experiments in interpretation guided segmentation. *Journal of Artificial Intelligence*, 8(3):241–274, 1977.

[34] R. Brooks, R. Greiner, and T. Binford. Model-based threedimensional interpretation of two-dimensional images. *In Proc. Int. Joint Conf. on Art. Intell.*, 1979.

[35] A. Hanson and E. Riseman. Visions: A computer system for interpreting scenes. *In Computer Vision Systems.*, 1978.

[36] Xuming He, Richard Zemel, and Deb Ray. Learning and incorporating top-down cues in image segmentation. In *European Conference on Computer Vision*. 2006.

[37] J. Shotton, J. Winn, C. Rother, and A. Criminisi. Textonboost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation. In *European Conference on Computer Vision*. 2006.

[38] Zhuowen Tu. Auto-context and its application to high-level vision tasks. In *IEEE Conference on Computer Vision Pattern Recognition*. 2008.

[39] Arpit Jain, Abhinav Gupta, and Larry S. Davis. Learning what and how of contextual models for scene labeling. In *European Conference on Computer Vision*. 2010.

[40] Li-Jia Li, Richard Socher, and Li Fei-Fei. Towards total scene understanding:classification, annotation and segmentation in an automatic framework. In *IEEE Conference on Computer Vision Pattern Recognition*. 2009.

[41] Lubor Ladicky, Chris Russell, Pushmeet Kohli, and Philip H. S. Torr. Associative hierarchical crfs for object class image segmentation. In *International Conference on Computer Vision*. 2009.

[42] Lubor Ladicky, Chris Russell, Pushmeet Kohli, and Philip H.S. Torr. Graph cut based inference with co-occurrence. In *European Conference on Computer Vision*. 2010.

[43] Tomasz Malisiewicz and Alexei A. Efros. Improving spatial support for objects via multiple segmentations. In *British Machine Vision Conference*. 2007.

[44] Eitan Sharon, Meirav Galun, Dahlia Sharon, Ronen Basri, and Achi Brandt. Hierarchy and adaptivity in segmenting visual scenes. *In the journal of Nature*, 442(7104):719–864, 2006.

[45] D. Martin, Fowlkes C., and J. Malik. Learning to detect natural image boundaries using local brightness, color, and texture cues. *IEEE Tran. on Pattern Analysis and Machine Intelligence*, 26:530–549, 2004.

[46] Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2011.

[47] D. Hoiem, A.A. Efros, and M. Hebert. Geometric context from a single image. In *International Conference on Computer Vision*. 2005.

[48] Prateek Jain and Ashish Kapoor. Active learning for large multi-class problems. In *IEEE Conference on Computer Vision Pattern Recognition*. 2009.

[49] Marius Leordeanu, Martial Hebert, and Rahul Sukthankar. An integer projected fixed point method for graph matching and map inference. In *Conference on Neural Information Processing Systems*. 2009.

[50] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A Zisserman. The pascal visual object classes challenge 2009 (voc2009) results.

[51] S. Satkin and M. Hebert. Modeling the temporal extent of actions. In *European Conference on Computer Vision*, 2010.

[52] Abhinav Gupta, Aniruddha Kembhavi, and Larry S. Davis. Observing human-object interactions: Using spatial and functional compatibility for recognition. *Pattern Analysis and Machine Intelligence*, 2009.

[53] Zhangzhang Si, Mingtao Pei, and Song-Chun Zhu. Unsupervised learning of event and-or grammar and semantics from video. In *International Conference on Computer Vision*, 2011.

[54] Vincent Delaitre, Josef Sivic, and Ivan Laptev. Learning person-object interactions for action recognition in still images. In *Conference on Neural Information Processing Systems*, 2011.

[55] Abhinav Gupta, Praveen Srinivasan, Jianbo Shi, and Larry S. Davis. Understanding videos, constructing plots: Learning a visually grounded storyline model from annotated videos. In *IEEE Conference on Computer Vision Pattern Recognition*, 2009.

[56] Pedro F. Felzenszwalb, Ross B. Girshick, David McAllester, and Deva Ramanan. Object detection with discriminatively trained part-based models. *Pattern Analysis and Machine Intelligence*, 2010.

[57] Lubomir Bourdev and Jitendra Malik. Poselets: Body part detectors trained using 3d human pose annotations. In *International Conference on Computer Vision*, 2009.

[58] Saurabh Singh, Abhinav Gupta, and Alexei A. Efros. Unsupervised discovery of mid-level discriminative patches. In *European Conference on Computer Vision*, 2012.

[59] J. Niebles, C.W. Chen, and L. Fei-Fei. Modeling temporal structure of decomposable motion segments for activity classification. In *European Conference on Computer Vision*, 2010.

[60] Alexei A. Efros, Alexander C. Berg, Greg Mori, and Jitendra Malik. Recognizing action at a distance. In *International Conference on Computer Vision*, 2003.

[61] Tomasz Malisiewicz, Abhinav Gupta, and Alexei A. Efros. Ensemble of exemplar-svms for object detection and beyond. In *International Conference on Computer Vision*, 2011.

[62] A.F. Bobick and J.W. Davis. The recognition of human movement using temporal templates. *Pattern Analysis and Machine Intelligence*, 2001.

[63] Lena Gorelick, Moshe Blank, Eli Shechtman, Michal Irani, and Ronen Basri. Actions as space-time shapes. In *International Conference on Computer Vision*, 2005.

[64] I. Laptev. On space-time interest points. *International Journal of Computer Vision*, 2005.

[65] Juan Carlos Niebles, Hongcheng Wang, and Li Fei-Fei. Unsupervised learning of human action categories using spatial-temporal words. *International Journal of Computer Vision*, 2008.

[66] Paul Scovanner, Saad Ali, and Mubarak Shah. A 3-dimensional sift descriptor and its application to action recognition. In *ACM Multimedia*, 2007.

[67] Heng Wang, Muhammad Muneeb Ullah, Alexander Klaser, Ivan Laptev, and Cordelia Schmid. Evaluation of local spatio-temporal features for action recognition. In *British Machine Vision Conference*, 2009.

[68] J. Liu, J. Luo, and M. Shah. Recognizing realistic actions from videos in the wild. In *IEEE Conference on Computer Vision Pattern Recognition*, 2009.

[69] A. Kovashka and K. Grauman. Learning a hierarchy of discriminative space-time neighborhood features for human action recognition. In *IEEE Conference on Computer Vision Pattern Recognition*, 2010.

[70] Bangpeng Yao, Xiaoye Jiang, Aditya Khosla, Andy Lai Lin, Leonidas J. Guibas, and Li Fei-Fei. Action recognition by learning bases of action attributes and parts. In *International Conference on Computer Vision*, 2011.

[71] Yan Ke, Rahul Sukthankar, and Martial Hebert. Event detection in crowded videos. In *International Conference on Computer Vision*, 2007.

[72] Y. Wang and G. Mori. Hidden part models for human action recognition: Probabilistic versus max margin. *Pattern Analysis and Machine Intelligence*, 2011.

[73] Tian Lan, Yang Wang, and Greg Mori. Discriminative figure-centric models for joint action localization and recognition. In *International Conference on Computer Vision*, 2011.

[74] Nataliya Shapovalova, Arash Vahdat, Kevin Cannons, Tian Lan, and Greg Mori. Similarity constrained latent support vector machine: An application to weakly supervised action classification. In *In European Conference on Computer Vision*, 2012.

[75] Michalis Raptis, Iasonas Kokkinos, and Stefano Soatto. Discovering discriminative action parts from mid-level video representations. In *IEEE Conference on Computer Vision Pattern Recognition*, 2012.

[76] Kevin Beyer, Jonathan Goldstein, Raghu Ramakrishnan, and Uri Shaft. When is "nearest neighbor" meaningful? In *International Conference on Database Theory*, 1999.

[77] Alexander Kläser, Marcin Marszałek, and Cordelia Schmid. A spatio-temporal descriptor based on 3d-gradients. In *British Machine Vision Conference*, 2008.

[78] Carl Doersch, Saurabh Singh, Abhinav Gupta, Josef Sivic, and Alexei A. Efros. What makes paris look like paris? *ACM Transactions on Graphics (SIGGRAPH)*, 2012.

[79] Li-Jia Li, Hao Su, Eric P. Xing, and L. Fei-Fei. Object bank: A high-level image representation for scene classification and semantic feature sparsification. In *Conference on Neural Information Processing Systems*, 2010.

[80] Ivan Laptev, Marcin Marszalek, Cordelia Schmid, and Benjamin Rozenfeld. Learning realistic human actions from movies. In *IEEE Conference on Computer Vision Pattern Recognition*, 2008.

[81] William Brendel and Sinisa Todorovic. Learning spatiotemporal graphs of human activities. In *International Conference on Computer Vision*, 2011.

[82] Adrien Gaidon, Zaid Harchaoui, and Cordelia Schmid. Recognizing activities with cluster-trees of tracklets. In *British Machine Vision Conference*, 2012.

[83] Ray Smith. An overview of the tesseract ocr engine. In *Proc. Ninth Int. Conference on Document Analysis and Recognition (ICDAR)*, 2007.

[84] ABBYY Mobile Products. Available: http://www.abbyy.com/mobile/accessed. 2013.

[85] Milyaev S, Barinova O, Novikova T, Lempitsky V, and Kohli P. Image binarization for end-to-end text understanding in natural images. In *International Conference on Document Analysis and Recognition (ICDAR)*, 2013.

[86] Keechul Jung, Kwang In Kim, and Anil K. Jain. Text information extraction in images and video: a survey. *Pattern Recognition*, 37(5):977–997, 2004.

[87] Serge Belongie Kai Wang, Boris Babenko. End-to-end scene text recognition. In *International Conference on Computer Vision*, 2011.

[88] Jung-Jin Lee, Pyong-Hean Lee, Seong-Whan Lee, Alan Yuille, and Christof Koch. Adaboost for text detection in natural scene. In *International Conference on Document Analysis and Recognition (ICDAR)*, 2011.

[89] Chucai Yi and Yingli Tian. Text detection in natural scene images by stroke gabor words. In *International Conference on Document Analysis and Recognition (ICDAR)*, 2011.

[90] Boris Epshtein, Eyal Ofek, and Yonatan Wexler. Detecting text in natural scenes with stroke width transform. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2011.

[91] Xujun Peng, Huaigu Cao, Rohit Prasad, and Premkumar Natarajan. Text extraction from video using conditional random fields. In *International Conference on Document Analysis and Recognition (ICDAR)*, 2011.

[92] Maximally Stable Extremal, J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide baseline stereo from. In *British Machine Vision Conference*, pages 384–393, 2002.

[93] L. Neumann and J. Matas. Real-time scene text localization and recognition. *IEEE Conference on Computer Vision and Pattern Recognition*, 0:3538–3545, 2012.

[94] Xuwang Yin, Xu-Cheng Yin, Hong-Wei Hao, and Khalid Iqbal. Effective text localization in natural scene images with mser, geometry-based grouping and adaboost. In *International Conference on Pattern Recognition*, 2012.

[95] Cunzhao Shi, Chunheng Wang, Baihua Xiao, Yang Zhang, and Song Gao. Scene text detection using graph model built upon maximally stable extremal regions. *Pattern Recognition Letters*, 34(2):107 – 116, 2013.

[96] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *IEEE Conference on Computer Vision Pattern Recognition*, 2005.

[97] C. Harris and M. Stephenes. A combined corner and edge detector. In *Alvey Vision Conference*, 1988.

[98] J. Canny. A computational approach to edge detection. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 8:679–714, 1986.

[99] H. Wold. Partial least squares. *In S. Kotz and N. Johnson, editors, Encyclopedia of Statistical Sciences*, 6:581–591, 1985.

[100] Nobuyuki Otsu. A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man and Cybernetics*, pages 62–66, 1979.

[101] Premkumar Natarajan, Zhidong Lu, Richard Schwatrz, Issam Bazzi, and John Makhoul. Multilingual machine printed ocr. *International Journal of Pattern Recognition and Artificial Intelligence*, 15, 2001.

[102] Paul Over, George Awad, Martial Michel, Jonathan Fiscus, Greg Sanders, Wessel Kraaij, Alan F. Smeaton, and Georges Quenot. Trecvid 2013 – an overview of the goals, tasks, data, evaluation mechanisms and metrics. In *Proceedings of TRECVID 2013*. NIST, USA, 2013.

[103] Carl Doersch, Abhinav Gupta, and Alexei A. Efros. Mid-level visual element discovery as discriminative mode seeking. In *Conference on Neural Information Processing Systems*, 2013.